Malmö högskola
Centrum för teknikstudier

Webbprogrammering, DA123A, Studiematerial 2

XML

XML – ”well formed” och ”valid”

XML står för *eXtensible Markup Language* och bygger precis som HTML på SGML och är även en rekommendation från W3C. XML är skapat för att strukturera information och underlätta utbyte av information mellan annars icke kompatibla plattformar. XML är ju ett markeringspråk men har inga fasta taggar utan används istället för att definiera andra markeringsspråk, så som exempelvis XHTML. XML innehåller ingen information hur data skall presenteras utan bara information om hur data är strukturerad och hur olika delar relaterar till varandra. Flera olika markeringsspråk har skapats i XML och för de mest skilda områden. Allt från kemi till tal finns representerat.

Precis som för HTML så är XML-filen i sig bara en textfil. XML skall vara väl formaterat(well formed) vilket innebär att exempelvis alla element måste stängas och att nästlade element måste stängas i samma ordning som de öppnats. Ett element består av en start-tag, sluttag och data mellan dessa. Kravet om formateringen säger dock ingenting om hur elementen skall struktureras. För detta används en DTD[1] (Document Type Definition) eller XML Schema[2].

Ett XML-dokument bör alltid börja med en XML-deklaration som anger version och teckenkodning.

```
<?xml version = ”1.0” encoding="ISO-8859-1"?>
```

Ett exempel på ett XML-dokument kan vara:

```
<?xml version = ”1.0” encoding="ISO-8859-1"?>
<!DOCTYPE bibliotek SYSTEM "Mitt_bibliotek.dtd">
<bibliotek>
  <bok typ=”barnbok”>
    <titel>Vi på Saltkråkan</titel>
    <författare>Astrid Lindgren</författare>
  </bok>
  <bok typ=”ungdomsbok”>
    <titel>Klockornas tid</titel>
    <författare>Maria Gripe</författare>
  </bok>
</bibliotek>
```

I exemplet ovan så är <bibliotek> det så kallade root-elementet. Alla andra element i ett dokument måste finnas inom root-elemetet. Som synes ovan så kan element även ha attribut.

Namn på element måste följa vissa regler:

- Namn kan innehålla bokstäver, siffror och andra tecken.
- Namn får inte starta med en siffra eller en punkt.
- Namn får inte starta med bokstavskombinationen xml.
- Namn kan inte innehålla blanktecken

[1] http://en.wikipedia.org/wiki/Document_Type_Definition , http://www.w3schools.com/dtd/default.asp
[2] http://en.wikipedia.org/wiki/XML_Schema_(W3C) , http://www.w3schools.com/schema/default.asp

För att ett XML-dokument skall vara ”well formed” så ska det följa följande regler:

- Det skall finnas ett root-element
- Alla element måste stängas med en slut-tag
- XML-taggar skiljer mellan gemener och versaler
- Elementen måste vara korrekt nästlade(stängas i samma ordning som de öppnades)
- Attribut måste alltid inneslutas i dubbla citationstecken.

För att ett XML-dokument skall vara giltigt (valid) så måste det även överensstämma med en DTD eller annat schema. Vi kommer här att titta närmare på DTD då detta används för att definiera hur HTML och XHTML får struktureras.

DTD

En DTD bestämmer hur ett XML-dokument får byggas upp, vilka taggar som finns och hur dessa får relatera till varandra. Byggstenarna i en DTD är:

- Elements – element
- Attributes – attribut
- Entities – enheter
- PCDATA
- CDATA

Entities är variabler som används för att definiera tecken på samma sätt som i HTML. & är en entity. Fördefinierade enheter i XML är

- < för <
- > för >
- & för &
- " för ”
- ' för ’

PCDATA står för *parsed character data*. Data av denna typ kommer att tolkas av en xml-parser. CDATA står för *character data* och kommer inte att tolkas av en xml-parser. DTDn Mitt_bibliotek.dtd för XML-exemplet ovan:

```
<!ELEMENT bibliotek (bok)>

<!ELEMENT bok(titel,författare)>
<!ATTLIST bok typ CDATA ”vuxenlitteratur”>

<!ELEMENT titel ( #PCDATA )>
<!ELEMENT författare ( #PCDATA )>
```

Mer och utförligare information om XML får du via de hänvisningar till länkar i fotnoter och länkar under rubriken Länkar senare i studiematerialet.

XHTML

Bakgrund

XHTML står för *eXtensible HyperText Markup Language* och är precis som HTML utvecklat av W3C. För den som kan HTML är det lätt att lära sig XHTML. Skillnaderna är inte så stora, men de är viktiga steg i utvecklingen mot integration och standardisering på Internet samt 3-lagers principen. Det finns många skäl till att använda XHTML istället för HTML. En del av skälen tas upp nedan. Det är också viktigt att uppmärksamma att det finns skäl till att inte alltid välja XHTML före HTML. I den kommande texten beskrivs även dessa skäl.

Varför är XHTML bättre än HTML?

HTML är som du bör veta utformat efter det generella markeringsspråket SGML. SGML är ett stort komplext system för att markera information. På grund av sin stora komplexitet har SGML inte använts så mycket till att direkt markera information, utan istället till att definiera nya enklare markeringsspråk. Ett nytt sådant markeringsspråk som definierats är XML. XML kan ses som en kompakt och striktare variant av SGML. Mycket av funktionaliteten finns kvar, men språket är striktare då man bland annat skiljer på gemener och versaler samt att alla taggar måste avslutas. XHTML är sedan utformat i XML och den första versionen som kom, kan ses som en översättning av HTML från SGML till XML.

Det kan tyckas märkligt att det är "enklare" att använda ett striktare markeringsspråk. Vi människor förstår oftast HTML även om sluttaggen för stycket saknas, eller om taggarna ömsom skrivs med versaler och ömsom med gemener. Lägg också märke till att det enligt HTML-standarden inte är fel att utelämna sluttaggen för stycke <p>, eller växla mellan gemener och versaler. Om däremot maskiner/datorer skall tolka informationen så är det mycket effektivare om markeringsspråket är så hårt definierat att man inte har någon valfrihet i dessa saker. Eftersom många tjänster i samhället har digitaliserats och många av dem även går att komma åt via Internet, så är det en stor fördel om maskiner/datorer enkelt och snabbt kan tolka och integrera information.

Ett annat viktigt steg mot en integrerad och standardiserad webb är att XHTML numera är fritt från de flesta formaterande taggar. Ända sedan senaste versionen av HTML (4.01), som kom 1999, har webbutvecklare uppmanats att skilja på information och presentation. För presentation används med fördel CSS, även det utvecklat av W3C. Arbetet med CSS version 3[3] pågår, men än så länge så är det version 1 och 2 som är de mest använda och också de som stöds av flest webbläsare.

Ny DTD

För XHTML-dokument krävs en ny DTD. Varje dokument inleds med en XML-deklaration och en deklaration om vilken XHTML-version som har använts. Man bör även specificera vilken teckenkodning som har använts. Det kan se ut till exempel som i exemplet nedan. Här används version 1.1 och den västerländska teckenkodningen ISO-8859-1 som går att använda i de svenskspråkiga varianterna av Windows, Linux, UNIX m.fl. DTD-deklaration för ett XHTML-dokument (XHTML 1.1) med västerländsk teckenuppsättning (ISO-8859-1) syns nedan.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11.dtd">
```

[3] http://www.w3.org/Style/CSS/current-work

Hur påverkar det koden i övrigt?
Vi börjar med att lista de mest uppenbara skillnaderna, det vill säga de som påverkar hur vi skriver våra dokument.

- Alla taggar måste skrivas med gemener (små bokstäver).
- Alla attribut måste skrivas med gemener.
- Alla taggar måste avslutas.
- Alla attributvärden måste skrivas innanför citattecken.

Att skriva med gemener lär man sig efter ett tag även om man är van att skriva taggarna med versaler. Händer det inget när du skriver en rubrik, eller fungerar inte attributet som du nyss skrev dit, så har du förmodligen skrivit med versaler.

I den andra punkten står det att alla taggar måste avslutas. För de slutna taggarna fungerar det precis som vanligt, men även de öppna taggarna måste nu avslutas. Det gör man genom att skriva in ett snedstreck i slutet av taggmarkeringen, se exemplet nedan. Ta som regel att även skriva ett mellanslag innan sista snedstrecket, på så sätt klarar även riktigt gamla webbläsare att förstå koden.

```
<p>Ett stycke i XHTML</p>
<!-- Även öppna taggar måste avslutas, som t.ex. bilden nedan -->
<img src="bild.jpg" />
<!-- Alla attributvärden måste omslutas med dubbla citattecken -->
<ol type="A">
   <li>En ordnad lista</li>
</ol>
```

Kodmässiga skillnader mellan HTML och XHTML
Utöver dessa skillnader så har från och med version 1.1 av XHTML stödet för de taggar som sedan version 1.0 och HTML 4.01 varit så kallade deprecated (ogillas, föråldrade) tagits bort. De taggar som blivit klassade som deprecated rekommenderas inte längre att användas. Dokument som använder dessa taggar räknas således inte som giltiga.

Utförligare beskrivning av skillnaderna hittar ni i specifikationen för XHTML 1.0. Vi avslutar med en mall på hur ett komplett (grunden) XHTML-dokument (för version 1.1, ISO-8859-1) utformas.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Titel</title>
  <meta http-equiv="Content-Type" content="application/xhtml+xml;
charset=iso-8859-1" />
  </head>
<body>
  <p>Innehållet skrivs här.</p>
</body>
</html>
```

Koden ovan finns även i exempelfilen ex_xhtml.xhtml. Lägg märke till meta-taggen som talar om vad det är för innehåll i filen.

IE klarar dock inte av att öppna denna fil då den inte vet hur den skall tolka ändelsen xhtml och inte heller riktigt klarar att hitta DTDn om man anger filen som en XML-fil. För filer som innehåller XHTML så får man för IE använda en filändelse som inte orsakar problem, exempelvis html. När du använder PHP senare i kursen, blir detta inget problem då XHTML genereras via en fil med ändelsen php. Koden ovan finns med olika filändelser till studiematerialet. Pröva att öppna filen via olika webbläsare och se vad som händer.

Validera din kod
För att säkerställa att ett dokument är korrekt utformat så finns en valideringstjänst hos W3C. Denna tjänst kan med fördel även användas för att lokalisera fel som trots timmars felsökning, fortfarande ihärdigt, vägrar att ge sig till känna. Valideringstjänsten finns här:
http://validator.w3.org/

Varför välja HTML istället för XHTML?
Trots att det talas varmt om XHTML i de flesta sammanhang, så finns det anledning till att ta en fundering på varför man inte vill använda "vanlig" HTML till sin webbsida. Om man använder valideringstjänsten (se ovan) och dessutom utformar sina dokument enligt HTML 4.01 Strict, så blir dokumenten nästan lika lätta att tolka som om de var skrivna i XHTML 1.0. Precis som XHTML så har alla deprecated och formaterande element och attribut tagits bort. DTD och dokumentstruktur för Strict HTML 4.01 kan ses i exemplet nedan:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
<html>
   <head>
      <title>Titel</title>
      <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
   </head>
   <body>
      <p>Innehållet skrivs här.</p>
   </body>
</html>
```

Om man skall välja XHTML så skall det vara på basis av att dokumenten är korrekt validerade och att det finns en mening med valet, till exempel att information skall integreras från någon annan XML-baserad teknik. Utan att göra en allt för omfattande analys om varför det kan uppstå oönskade fel om man använder XHTML på fel sätt, så görs en kort sammanfattning i följande lista.

- Ett CSS dokument tolkas lite annorlunda i XHTML. T.ex. får inte versaler användas.
- <script> och <style> är inte definierade på riktigt samma sätt och kan ställa till problem.
- Script som använder DOM fungerar på ett lite annorlunda sätt i XHTML.
- Funktionen document.write() fungerar inte i XHTML (ävan om man av andra orsaker inte vill använda denna som vi kommer att se i senare studiematerial).
- Skapande av nya element med DOM görs inte på riktigt samma sätt.

Skrivsättet med lite annorlunda och inte riktigt samma i listan ovan kan tyckas tyda på att problemen inte är så allvarliga. Men det är just dessa små skillnader som gör det extra vanskligt, det kan verka som om det visst fungerar i XHTML. Det som troligtvis då har hänt är att webbläsaren har tolkat koden som HTML istället, vilket gör att den dagen som

webbläsare verkligen tolkar dokumentet "rätt" så kan webbsidan/webbplatsen komma att haverera fullständigt.

För en mer utförlig beskrivning om varför XHTML bör väljas med omsorg, se http://www.hixie.ch/advocacy/xhtml. Där finns även en referens till länken: http://www.xml.com/pub/a/2003/03/19/dive-into-xml.html, som är lite lättare att ta till sig.

Åsikterna om XHTML/HTML och Mime-type går dock isär på flera områden. En artikel som skildrar problematiken på ett annat sätt är: http://www.sitepoint.com/article/html-or-xhtml-does-it-matter .

Så i slutänden så måste utvecklaren själv avgöra vad som skall användas och hur och motivera det val som görs.

Vad skall vi använda i kursen?
I kursen kommer vi att använda båda teknikerna. Vi kommer främst att använda HTML när vi sysslar med JavaScript och DHTML, för att sedan använda XHTML till viss del i PHP-delen. Vi är medvetna om att de webbläsare som används kanske inte klarar MIME typen application/xhtml+xml alls eller ger parserproblemen. Men då XML-parsingen kommer att fungera på sikt och XHTML vinner allt mer mark i samband med att XML-applikationerna blir allt fler så vill vi att ni skall ha prövat på att använda i XHTML.

Länkar

XML
Sida 1-4 ska läsas: http://www.sitepoint.com/article/introduction-xml
Vissa exempel är kanske svåra att förstå än så länge men texten har ni nytta av.

Andra intressanta sidor:
http://www.w3c.se/resources/office/translations/XML-in-10-points_sw.html
http://en.wikipedia.org/wiki/Document_Type_Definition
http://en.wikipedia.org/wiki/XML_Schema
http://www.w3schools.com/xml/default.asp
http://www.w3schools.com/dtd/default.asp

XHTML vs. HTML
Ska läsas: http://www.hixie.ch/advocacy/xhtml
Alternativt: http://www.xml.com/pub/a/2003/03/19/dive-into-xml.html

Ska läsas: http://www.sitepoint.com/article/html-or-xhtml-does-it-matter

Validering
Du kommer att använda denna valideringstjänst på kursen: http://validator.w3.org/

Standarder
http://www.w3.org/TR/xhtml11/
http://www.w3.org/TR/xhtml1/
http://www.w3.org/TR/html4/
http://www.w3.org/TR/2000/REC-xml-20001006

Dessa är kanske inte så lätta att ta till sig men du bör känna till var de finns.

Fortsättning…

Vi tar här bara ytligt upp XML så ni känner till vad det är och ungefär hur det fungerar. Det praktiska syftet är att ni skall kunna läsa en DTD för att ta reda på vad som gäller för HTML eller XHTML. För den som är ytterligare intresserad av XML så rekommenderas en kurs som enbart sysslar med detta.

När det gäller XHTML och dess framtid och hur webbläsare i nya versioner kommer att hantera MIME-typen application/xhtml+xml så kan vi bara hålla ögonen på det som kommer och se vad som händer.