



Høgskulen
på Vestlandet

SAMSPEL
BEREKRAFT
NYSKAPING

Databaser

XML, DTD og XML Schema (XSD)

Dag Larsen
Oppdatert av Lasse Jenssen



Plan for NoSQL delen

- › Video 1: Introduksjon til NoSQL
- › XML og JSON (Kap. 14)
 - **Video 2: XML, DTD og XML Schema**
 - Video 3: DOM, XSLT, XPath, XQuery, XML i PostgreSQL
 - Video 4: JSON, JSON i PostgreSQL
- › Via objekter (objekt relasjonelle databasesystemer) til NoSQL databaser (Kap. 15)
 - Video 5: Via objekter til NoSQL databaser
 - Video 6: NoSQL databaser
- › NoSQL og MongoDB (Kap. 15.4)
 - Diverse videoer (kommer tilbake til disse etter hvert)
 - Vi bruker Docker eller MongoDB Atlas (eventuelt lokal installasjon på PC)
- › Obligatorisk Innlevering 4 (legges ut i starten av uke 15)

Læringsmål

- › Kunne lage **XML**-dokumenter med **elementer** og **attributter**.
- › Kunne beskrive lovlig oppbygging av XML-dokumenter med **DTD** og **XML Schema**.

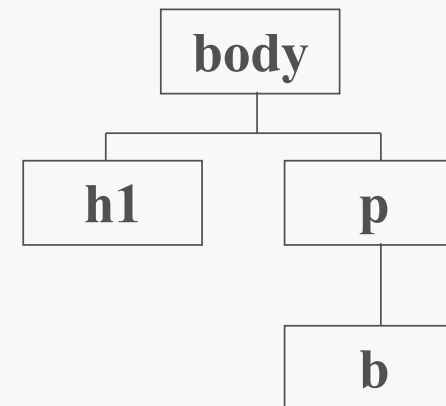
Oppbygging av XML-dokumenter

- › XML har «HTML-syntaks»

```
<?xml version="1.0" encoding="UTF-8"?>  
<melding dato="07.12.2019" rom="5-116">  
  <avsender fnavn="Kari" enavn="Lie"/>  
  <beskjed>Møte om 5 min!</beskjed>  
</melding>
```

HTML vs XML - I

```
<body>
  <h1>HTML</h1>
  <p>
    <b>HTML</b> er et språk
    for å beskrive nettsider.
  </p>
</body>
```

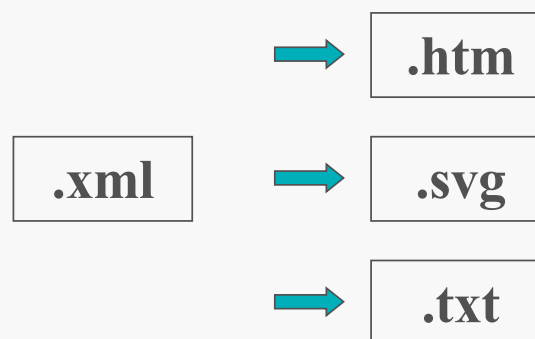


- › HTML er bygd opp fra elementer i en tre-struktur
- › Lovlige HTML-elementer er forhåndsdefinert
 - › Nettlesere vet hva **h1**, **p** og **b** betyr
- › HTML beskriver både **struktur** og **presentasjon**

HTML vs XML - II

- › XML er **utvidbart** (extensible)
 - › Ingen forhåndsdefinerte elementer
 - › Må tolkes for å bli presentert

```
<sjakktrekk>
  <brikke>
    <farge>hvit</farge>
    <type>tårn</type>
  </brikke>
  <fra>D1</fra>
  <til>D8</til>
</sjakktrekk>
```



- › Kan vises som teksten **Hvitt tårn fra D1 til D8**, eller som en animasjon.

XML - eXtensible Markup Language

- › Syntaksregler for **velformet** XML
 - › Hvert XML-dokument har ett og bare ett **rot-element**
 - › Alle elementer har både start-tag og slutt-tag
 - › Hvert element kan ha et antall attributter (ikke to attributter med samme navn)
 - › Kan ha nøstede elementer som i HTML, men elementer kan ikke overlappe
- › XML er et **metaspråk** = språk for å definere nye språk
 - › Finnes for mange fagområder og bransjer, f.eks:

Geografiske data (GML)	www.opengis.net/gml/
e-handel (ebXML)	www.ebxml.org
Kjemiske formler (CML)	www.xml-cml.org

Navnerom

- › Navnerom er innført for å unngå navnekollisjoner.
- › Kan ha to elementer med samme navn så lenge de er definert i forskjellige navnerom.

```
<studium xmlns = "http://www.hvl.no/kurs/"  
  xmlns:k = "http://www.hvl.no/skipsfart/"  
  <k:kurs>Nordvest</k:kurs>  
  <kurs>DAT107 - Databaser<kurs> ...
```

- › URL'er (URI) brukes for unik navngiving, og trenger ikke å være en nettsadresse som eksisterer.

DTD (Document Type Definition)

- › DTD beskriver lovlig oppbygging av XML-dokumenter:

```
<!ELEMENT FILM (ENFILM)*>
```

```
<!ELEMENT ENFILM (TITTEL, SJANGER?, PRODÅR>
```

```
<!ELEMENT TITTEL (#PCDATA)
```

```
<!ELEMENT SJANGER (#PCDATA)
```

```
<!ATTLIST ENFILM FilmNr CDATA #REQUIRED>
```

```
<!ATTLIST ENFILM Sensur CDATA #IMPLIED>
```

- › Et XML-dokument som er «syntaktisk korrekt» sies å være **velformet**.
- › Et XML-dokument som (dessuten) tilfredsstiller en DTD er **gyldig**.

https://www.tutorialspoint.com/dtd/dtd_quick_guide.htm

XML Schema


- › Også kalt «XML Schema Definition»: XSD
- › Strukturen til et XML-skjema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.hvl.no"
  xmlns="http://www.hvl.no"
  elementFormDefault="qualified">
  <!-- Typedefinisjoner -->
  <!-- Elementdefinisjoner -->
</xs:schema>
```

XML Schema

- › Også kalt «XML Schema Definition»: XSD
- › Strukturen til et XML-skjema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.hvl.no"
  xmlns="http://www.hvl.no"
  elementFormDefault="qualified">
  <!-- Typedefinisjoner -->
  <!-- Elementdefinisjoner -->
</xs:schema>
```

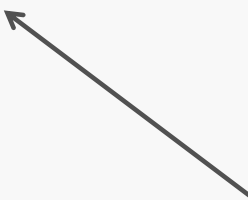


Elementer og typer
som brukes i schema
definisjon kommer
fra dette namespace.

XML Schema

- › Også kalt «XML Schema Definition»: XSD
- › Strukturen til et XML-skjema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.hvl.no"
  xmlns="http://www.hvl.no"
  elementFormDefault="qualified">
  <!-- Typedefinisjoner -->
  <!-- Elementdefinisjoner -->
</xs:schema>
```



Elementer og typer
definert av dette
schemaet kommer fra
(eller tilhører)
dette namespace.

XML Schema

- › Også kalt «XML Schema Definition»: XSD
- › Strukturen til et XML-skjema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.hvl.no"
  xmlns="http://www.hvl.no"
  elementFormDefault="qualified">
  <!-- Typedefinisjoner -->
  <!-- Elementdefinisjoner -->
</xs:schema>
```

Elementer som blir brukt i XML dokumenter som er deklartert i dette skjemaet må namespace kvalifiseres.

Koble XML og XML Schema

- › Opprett 2 filer på (f.eks.) www.xyz.no:
 - › personer.xml
 - › personer.xsd
- › XML-filen (personer.xml) refererer til XML-skjemaet :

```
<?xml version="1.0" encoding="utf-8"?>
<personer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="personer.xsd" >
  <person> ... </person>
  <person> ... </person>
</personer>
```

<https://www.w3.org/2001/XMLSchema>

Koble XML og XML Schema

- › Opprett 2 filer på (f.eks.) www.xyz.no:
 - › personer.xml
 - › personer.xsd
- › XML-filen (personer.xml) refererer til XML-skjemaet :

```
<?xml version="1.0" encoding="utf-8"?>
<personer xmlns="http://home.hvl.no/databaser"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://home.hvl.no/databaser
                              personer.xsd">

  <person> ... </person>
  <person> ... </person>
</personer>
```

<https://www.w3.org/2001/XMLSchema>

Innebygde datatyper

- › Et element kan tildeles en av de innebygde datatypene:

- › string
- › decimal
- › integer
- › boolean
- › date
- › time

Eksempel på XML-data:

```
<fornavn>Per</fornavn>  
<barn>2</barn>  
<fødselsdato>1994-02-27</fødselsdato>
```

- › Eksempler:

```
<xs:element name="fornavn" type="xs:string"/>  
<xs:element name="barn" type="xs:integer"/>  
<xs:element name="fødselsdato" type="xs:date"/>
```

- › Prefikser som regel med navnerom (xs må deklarereres)

Enkle datatyper ved restriksjon: Lengde

- › Kan lage egne (enkle) datatyper ved restriksjon av de innebygde.

- › Restriksjon på **antall tegn** (lengde)

```
<xs:simpleType name="fornavnType">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="20"/>  
  </xs:restriction>  
</xs:simpleType>
```

Gir et element denne datatypen:

```
<xs:element name="fornavn" type="fornavnType"/>
```

Enkle datatyper ved restriksjon: Intervall

- › Restriksjon på **verdiområde**:

```
<xs:simpleType name="pnrType">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="1"/>  
    <xs:maxInclusive value="500"/>  
  </xs:restriction>  
</xs:simpleType>
```

Gir et element denne datatypen:

```
<xs:element name="pnr" type="pnrType"/>
```

Enkle datatyper ved restriksjon: Mønster

- › Et **mønster** er et slags regulært uttrykk:

```
<xs:simpleType name="regnrType">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[A-Z]{2}[0-9]{5}"/>  
  </xs:restriction>  
</xs:simpleType>
```

Gir et element denne datatypen:

```
<xs:element name="regnr" type="regnrType"/>
```

Komplekse datatyper

- › En kompleks datatype kan f.eks. modellere rader i en tabell:

```
<xs:complexType name="personType">  
  <xs:sequence>  
    <xs:element name="pnr" type="pnrType"/>  
    <xs:element name="fornavn" type="fornavnType"/>  
    <xs:element name="etternavn" type="xs:string"/>  
    <xs:element name="tlf" type="telefonType" />  
  </xs:sequence>  
</xs:complexType>
```

- › Kan bruke både innebygde og egendefinerte datatyper når vi definerer en ny kompleks datatype

Antall forekomster

- › Vi kan definere at en type kan inneholde fra 0 til vilkårlig mange forekomster:

```
<xs:complexType name="personerType">  
  <xs:sequence>  
    <xs:element name="person" type="personType"  
      minOccurs="0" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

- › Merk for øvrig at personerType refererer til personType

```
<xs:element name="personer" type="personerType"/>
```

Attributter

- › Hvis pnr er et attributt:

```
<person pnr="1" >  
  <fornavn>Per</fornavn>  
</person>
```

- › Kan tilordne attributtet en datatype:

```
<xs:complexType name="personType">  
  <xs:sequence>  
    <xs:element name="fornavn" type="fornavnType"/>  
    ...  
  </xs:sequence>  
  <xs:attribute name="pnr" type="xs:int" use="required" />  
</xs:complexType>
```

- › Kan også bruke egendefinerte datatyper her.

Attributter

```
<person id="1" personnr="04117241314" fornavn="Ola"  
etternavn="Hansen" barn="2" fodselsdato="1991-10-02">  
</person>
```



```
<person id="1">  
  <personnr>04117241314</personnr>  
  <fornavn>Ola</fornavn>  
  <etternavn>Hansen</etternavn>  
  <barn>2</barn>  
  <fodselsdato>1991-10-02</fodselsdato>  
</person>
```



Unique attributt

- › Hvis pnr er et attributt:

```
<person pnr="1" >  
  <fornavn>Per</fornavn>  
</person>
```

- › Kan tilordne unik egenskap:

```
<xs:element name="personer" type="personerType">  
  <xs:unique name="pnrUnique" >  
    <xs:selector xpath="dat:person"/>  
    <xs:field xpath="@pnr"/>  
  </xs:unique>  
</xs:element>
```


DTD vs XML Schema

- › Svakheter med DTD:
 - › Ingen datatyper, kun tekststrenger
 - › Svak støtte for primærnøkler, fremmednøkler og valideringsregler
 - › Særegen syntaks, ikke XML-basert
- › XML Schema = en rikere DTD
 - › Kan benyttes som datadefinisjonsspråk for XML-dokumenter
 - › Flere datatyper, både enkle og brukerdefinerte
 - › Primærnøkler, fremmednøkler og valideringsregler
 - › Bruker XML-syntaks
 - › Støtter XML navnerom

Oppsummering

- ❑ XML = eXtensible Markup Language
 - Elementer og attributter
 - Prolog, navnerom, prosesseringsinstruksjoner

- ❑ Beskrive **velformulert** vs. **gyldig** XML: DTD, XML Schema