

XSLT++

Umbraco DK Festival 2010

Chriztian Steinmeier

Mig, mig, mig...

- Frontend Developer hos Heyday i Århus

Mig, mig, mig...

- Frontend Developer hos Heyday i Århus
- Masser af XSLT erfaring (10+ år)
- Halvandet års Umbraco erfaring

Det "nye" XML Schema

Det "nye" XML Schema

Document Typer:

```
<node id="2812" nodeTypeAlias="NewsItem" />  
...  
<NewsItem id="2812" isDoc="" />
```

XML

Egenskaber:

```
<data alias="umbracoNaviHide">1</data>  
...  
<umbracoNaviHide>1</umbracoNaviHide>
```

XML

Det "nye" XML Schema

Før:

```
<xsl:value-of select="node[not(data[@alias = 'umbracoNaviHide'] = 1)]" />
```

XSLT

Nu:

```
<xsl:value-of select="NewsItem[not(umbracoNaviHide = 1)]" />
```

XSLT

Demo

Templates (recap)

Templates

- Man laver en template til et element...

```
<xsl:template match="NewsItem" >  
  <!-- do stuff -->  
</xsl:template>
```

XSLT

Templates

- Fylder det ønskede output i...

```
<xsl:template match="NewsItem">
```

```
  <article>
```

```
    <header>
```

```
      <h1>Fisk fundet i live</h1>
```

```
    </header>
```

```
    <p> Den fisk der i går blev meldt savnet[...] </p>
```

```
  </article>
```

```
</xsl:template>
```

XSLT

Templates

- Indsæt det ønskede indhold fra XML dokumentet

```
<xsl:template match="NewsItem">
  <article>
    <header>
      <h1>
        <xsl:value-of select="newsHeader" />
      </h1>
    </header>
    <p>
      <xsl:value-of select="bodyText" disable-output-escaping="yes" />
    </p>
  </article>
</xsl:template>
```

XSLT

Templates

“hokus pokus...”

Templates

Fisk fundet i live!

Den fisk der i går blev meldt savnet af lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.

Tallene er onde

Der florerer informationer om en speciel dolor amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.

XSLT++

XSLT++

- *xsl:if* og *xsl:choose*

XSLT++

- *xsl:if* og *xsl:choose*
- *xsl:param* og *xsl:variable*

XSLT++

- *xsl:if* og *xsl:choose*
- *xsl:param* og *xsl:variable*
- *Media*

XSLT++

- *xsl:if* og *xsl:choose*
- *xsl:param* og *xsl:variable*
- *Media*
- *(Bonus?)*

Væk med den "if"...

Et ofte set pattern:

```
<xsl:template match="/">  
  <xsl:if test="string($currentPage/pageBanner) != ' ' ">  
    <!-- do stuff -->  
  </xsl:if>  
</xsl:template>
```

XSLT

Væk med den "if"...

Også kendt som:

```
<xsl:template match="/">  
  <xsl:if test="string-length($currentPage/pageBanner) > 0" >  
    <!-- do stuff -->  
  </xsl:if>  
</xsl:template>
```

XSLT

Væk med den "if" ...

Brug template:

```
<xsl:template match="/">
  <xsl:apply-templates select="$currentPage/pageBanner[normalize-space()]" />
</xsl:template>
<xsl:template match="pageBanner" >
  <!-- do stuff -->
</xsl:template>
```

XSLT

Ud med choose

Pattern:

```
<xsl:template match="/">
  <xsl:choose>
    <xsl:when test="$sortdir = 'op'">
      <xsl:apply-templates select="NewsItem">
        <xsl:sort order="ascending" />
      </xsl:apply-templates>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select="NewsItem">
        <xsl:sort order="descending" />
      </xsl:apply-templates>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

XSLT

Ud med choose

```
<xsl:template match="/">
  <xsl:apply-templates select="NewsItem[$sortdir = 'op']" >
    <xsl:sort order="ascending" />
  </xsl:apply-templates>
  <xsl:apply-templates select="NewsItem[$sortdir = 'ned']" >
    <xsl:sort order="descending" />
  </xsl:apply-templates>
</xsl:template>
```

XSLT

Ud med choose

Også til forskellige modes:

```
<xsl:template match="/">  
  <xsl:apply-templates select="Event[$mode = 'oversigt']" mode="oversigt" />  
  <xsl:apply-templates select="Event[$mode = 'detaljer']" mode="detaljer" />  
</xsl:template>
```

XSLT

"Variabler"

"Variabler"

"LOL"

Variabler

```
<xsl:variable name="myCoolVarName" select="expression" />
<!-- Alt det andet man laver -->
<xsl:template match="/">
  <xsl:value-of select="$myCoolVarName" />
</xsl:template>
```

XSLT

Variabler

Tal, strenge & set:

```
<!-- Antal retter pr. side: -->  
<xsl:variable name="antalRetter" select=" 25 " />  
<!-- Prefix til Dagens Ret: -->  
<xsl:variable name="todaysSpecialPrefix" select=" 'Dagens Ret: ' " />  
<!-- Hvor ligger de? -->  
<xsl:variable name="opskriftMappe" select="$siteRoot/Opskrifter" />  
<!-- Dem der skal vises: -->  
<xsl:variable name="$opskrifter" select="$opskriftMappe/Opskrift" />
```

XSLT

Default værdier til parametre

Pattern:

XSLT

```
<xsl:call-template name="build-options">
  <xsl:with-param name="from" select="1" />
  <xsl:with-param name="to" select="10" />
</xsl:call-template>

<xsl:template name="build-options">
  <xsl:param name="from" />
  <xsl:param name="to" />
  <!-- do stuff -->
  <xsl:if test="not($from = $to)">
    <xsl:call-template name="build-options">
      <xsl:with-param name="from" select="$from + 1" />
      <xsl:with-param name="to" select="$to" />
    </xsl:call-template>
  </xsl:if>
</xsl:template>
```


Brug defaults:

```
<xsl:call-template name="build-options" />
```

XSLT

```
<xsl:template name="build-options">
  <xsl:param name="from" select="1" />
  <xsl:param name="to" select="10" />
  <!-- do stuff -->
  <xsl:if test="not($from = $to)">
    <xsl:call-template name="build-options">
      <xsl:with-param name="from" select="$from + 1" />
    </xsl:call-template>
  </xsl:if>
</xsl:template>
```

“Crazy Cool Wacky” version:

```
<xsl:call-template name="build-options" />

<xsl:template name="build-options">
  <xsl:param name="from" select="1" />
  <xsl:param name="to" select="10" />
  <xsl:for-each select="document('')//*[position() <= (($to - $from) + 1)]">
    <!-- do stuff -->
  </xsl:for-each>
</xsl:template>
```

XSLT

Media

Media

Typisk snippet:

```
<xsl:for-each select="$opskrifter">
  <xsl:if test="billede != ''">
    
  </xsl:if>
</xsl:for-each>
```

XSLT

Media

Brug variable:

```
<xsl:for-each select="$opskrifter">  
  <xsl:variable name="mediaNode" select="umb:GetMedia(billede, false())" />  
    
</xsl:for-each>
```

XSLT

Media

Safeguard vha. template:

```
<xsl:for-each select="$opskrifter">
  <xsl:variable name="mediaNode" select="umb:GetMedia(billede, false())" />
  <xsl:apply-templates select="$mediaNode[not(error)]" />
</xsl:for-each>
<!-- kode, kode, kode -->
<xsl:template match="Image">
  
</xsl:template>
```

XSLT

_MediaHelper.xslt

Demo

Ses vi?

XSLT Ping Pong Fest i morgen!

TAK!

- **Twitter:** [@greystate](#)
- **Blog:** <http://greystate.dk>
- **XSLT 'pimpage':** <http://pimpmyxslt.com>
- **Work:** <http://www.heyday.dk>

Credits

- Awesomeness: our.umbraco.org, Umbraco + Hartvig & Co.
- Custom font: “planet e-style” by Mads Rydahl