# Oefeningen Xml Technologie Les 1

Surf eerst naar

<http://free.editix.com/>

en download en installeer de free version van EditiX, de xml editor die we gaan gebruiken in de les.

Voor de oefeningen die volgen, zijn deze 2 buttons heel belangrijk :



Check XML Syntax

Format XML

Met de eerste button kan je controleren of de xml die je hebt gemaakt geldige xml is. De editor geeft gerichte foutboodschappen als dat niet het geval is. Met de tweede button kan je de layout van je xml automatisch laten indenteren of inspringen, zodat je de hiërarchische structuur beter ziet.

Als je nog niet werkt met NotePad++, download en installeer deze nuttige editor dan ook :

<http://notepad-plus-plus.org/download/v5.9.8.html>

1. Zoek uit hoe je een nieuw standaard xml document maakt. De editor zet daar al het volgende in klaar :

<?xml version="1.0" encoding="UTF-8"?>

En een stuk commentaar :

<!-- New document created with EditiX at Tue Jan 31 11:30:09 CET 2012 -->

Negeer deze 2 lijnen, laat ze gewoon staan. Uitleg volgt later.

1. Maak nu de xml structuur die in de volgende specs wordt gedefiniëerd. Het is de bedoeling dat je de xml zelf intypt in EditiX, te beginnen op de lijn na de commentaar. Tekst die in italic staat mag je copy/pasten.

Deze xml bevat de inhoud van een kort document over een onderdeel van Artificial Intelligence, namelijk *textmining*.

Omdat de xml een beschrijving is van een document, gebruik je als root (dat is het eerste element) het element ‘document’. Het document begint met een header als eerste child element. Dit element heeft zelf 2 child elementen. De tekst van de header moet gecentreerd worden. Gebruik hiervoor het element ‘justify’, met als content ‘center’. De tekst zelf is het tweede child element, met als content :

*Eerste Oefening op XML*

Op hetzelfde niveau als ‘header’ zet je de title. Hiervoor moet je enkel de tekst opgeven in een element :

*Text mining*

Dan volgt de eerste paragraph, die een attribuut heeft, ‘number’ met waarde 1, en een element ‘text’ :

*Text mining is een interdisciplinair vakgebied dat tot doel heeft technieken en methoden te ontwikkelen om kennis uit vrije tekst te extraheren. De disciplines die zich in text mining verenigen zijn onder meer enkele vakgebieden uit de informatica/kunstmatige intelligentie zoals information retrieval en machine learning, maar ook disciplines als statistiek en computationele taalkunde.*

Dan volgt de tweede paragraph, volledig analoog als de eerste, maar met een andere waarde voor number natuurlijk :

*Information retrieval (IR) houdt zich bezig met het zoeken naar informatie in documenten, naar documenten zelf, naar metadata die de documenten beschrijft, en het zoeken binnen databases, naar tekst, audio, beelden, of data. De termen data retrieval, document retrieval, information retrieval, en text retrieval worden vaak door elkaar gebruikt, hoewel ze elk over eigen literatuur, theorie, praktijk, en technologieën beschikken.*

Tenslotte volgt er een footer waarvoor je het element ‘justify’ op left zet, en de volgende tekst gebruikt :

*Textmining is een onderdeel van Artificial Intelligence*

Kijk na of de xml die je hebt gemaakt geldig is, en save als ‘textmining.xml’.

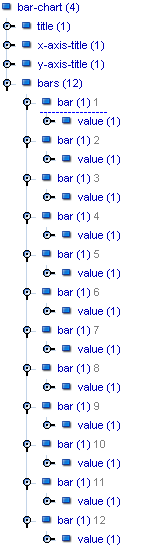
1. Doe hetzelfde voor de volgende specs. Hier wordt meer met attributen gewerkt.

Deze xml bevat cijfergegevens die in een *barchart* moeten worden weergegeven. Dat wordt dus de root. De barchart zelf heeft een title (*Bar Chart: Lengths*), en ook de X en de Y as hebben een title (*Items* en *Lengths*).

Er zijn 12 metingen uitgevoerd. Die komen in het element ‘bars’ dat per meting een element ‘bar’ heeft. Elke bar heeft een attribuut ‘number’, dat loopt van 1 tot en met 12. Elke bar heeft een element ‘value’ dat de gemeten waarde bevat. Gebruik hiervoor de volgende waarden :

90, 40, 130, 140, 120, 60, 70, 80, 120, 100, 25, 20

De xml tree ziet er uit zoals op de volgende pagina wordt weergegeven :



Kijk na of de xml die je hebt gemaakt geldig is, en save als ‘bars.xml’.

1. Doe hetzelfde voor de volgende specs die uit de xml taal komen om muziek voor te stellen, MusicXml. Dit is een moeilijkere oefening.

* Open de file *simple music.xml* . De algemene structuur is al voorzien. Na het element *part* staat er in commentaar dat je op die plaats de oefening moet maken.
* Een muziekstuk wordt een *part* genoemd. Een part heeft een attribuut met naam ‘id’ en waarde *P1*. Dat staat er al.
* Elk part bestaat uit muziekmaten of measures, in deze oefening is dat er 1. Het element ‘measure’ heeft een attribuut met naam ‘number’ en waarde het nummer van die maat.
* Elke measure bevat een element ‘attributes’ en een aantal elementen ‘note’.
* Het element ‘attributes’ heeft de volgende structuur :
* Een element ‘divisions’ met als content 1.
* Een element ‘key’ met als content een element ‘fifths’ dat als content 0 heeft.
* Een element ‘time’ dat 2 child elementen heeft : ‘beats’ met als content 4 en ‘beat-type’ met als content 4.
* Een element ‘clef’ dat 2 child elementen heeft : ‘sign’ met als content ‘G’ en ‘line’ met als content 2.
* Elke note heeft de volgende structuur :
* Een element ‘pitch’ dat 2 child elementen heeft : ‘step’ en ‘octave’.
* Een element ‘duration’.
* Een element ‘type’.

Maak nu 4 noten aan op de juiste plaats, met de volgende waarden :

Noot 1

step = C

octave = 4

duration = 2

type = quarter

Noot 2

step = E

octave = 4

duration = 2

type = quarter

Noot 3

step = G

octave = 4

duration = 2

type = quarter

Noot 4

step = C

octave = 5

duration = 2

type = quarter

Kijk na of de xml die je hebt gemaakt geldig is, en save als ‘music.xml’.

Installeer nu het bijgeleverde programma MusicEase (*musezw.exe*), dat is een simpele musicxml player. Open het bestand en geef het weer. Je zou het volgende moeten zien :



1. Doe hetzelfde voor de volgende specs. Hier worden geen attributen gebruikt. Dit is een toepassing van MathMl. Deze taal wordt wereldwijd gebruikt om wiskundige formules weer te geven.

De xml die je hier moet maken stelt de volgende formule voor : **(a + b)2**. Het uiteindelijke resultaat zal er in xml veel complexer uitzien, maar wel door alle MathMl processors van de wereld begrepen worden.

Elke tag in MathMl begint met de letter m.

* De hele expressie wordt als row bekeken. De root zal dus ‘mrow’ heten.
* De root bevat 1 element, ‘msup’ van superscript, omdat de expressie een macht bevat.
* ‘msup’ bevat 2 elementen die je kan bezien als argumenten van een method : (a + b) en de macht tot 2 zelf.
* Eerste element : *tussen haakjes* is in het Engels *fenced*. Omdat a + b tussen haakjes staat, heet dit element ‘mfenced’.
* Tweede element : dit is de macht tot 2 zelf en wordt getagd als ‘mn’. De letter n staat voor number.
* We hebben nu 2 haakjes en een macht. We hebben nog niets gezegd over de som zelf. Die wordt ook als row beschouwd. De tag ‘mfenced’ krijgt dus als enige element een ‘mrow’.
* Die ‘mrow’ bevat 3 elementen, twee voor de identifiers a en b, en één voor de operator +. Die heten ‘mi’ en ‘mo’.

Kijk na of de xml die je hebt gemaakt geldig is, en save als ‘expression.xml’. Kijk ook goed na of de logica wel klopt. Als je door de xml leest, moet je de formule terug kunnen vinden.

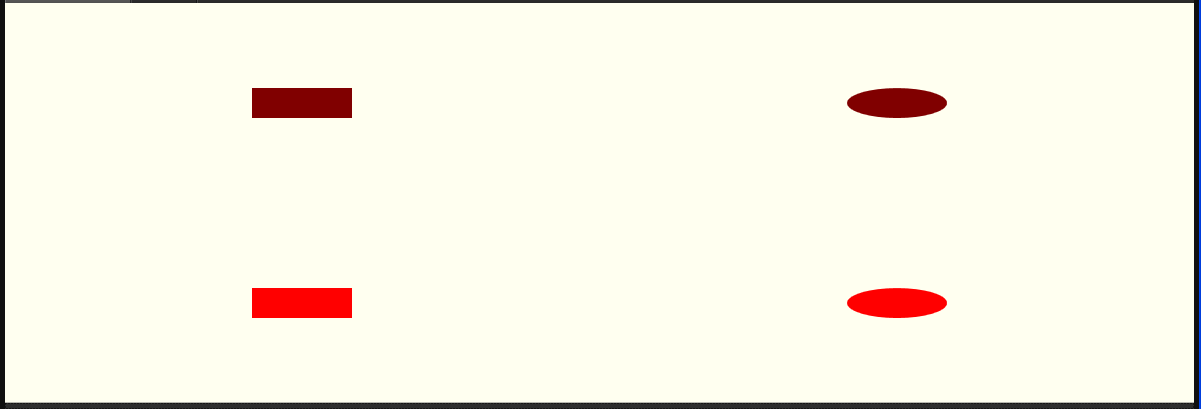
Ga nu naar <http://www.mathmlcentral.com/Tools/FromMathML.jsp>. Verwijder het <msup> element dat er staat en paste daarover jouw resultaat. Check het resultaat.

1. In deze oefening maken we de eerste xaml file.

Vooraf :

* Kijk na in configuratiescherm, bij Software, of er iets geïnstalleerd is van Silverlight. Als dat zo is, verwijder dan alles.
* Surf naar <http://www.microsoft.com/getsilverlight/Get-Started/Install/Default.aspx> en installeer Silverlight 5. Dit is de plugin die er voor zorgt dat je xaml files kan bekijken in je browser, vergelijkbaar met Flash.
* Ga naar <http://www.kaxaml.com/> en download en installeer KaXaml.exe. Dit is de xaml editor die we gaan gebruiken.
* Open de file *grid.xaml* in KaXaml. Open de file daarna met Internet Explorer. Als dat allemaal lukt en je kan de file in beide omgevingen bekijken, kan je verder.

Maak nu een xaml file volgens de specs die volgen. Xaml is hoofdlettergevoelig ! Het resultaat zou moeten zijn :



* Open de file *first empty.xaml* in KaXaml.
* De root van het xml document staat er al, *Page*. In Page worden 2 namespaces opgegeven als attribuut. Stel je daar nog niet te veel vragen bij, maar de root moet er zo uitzien, anders kan KaXaml niets weergeven. De uitleg hierover volgt later.
* De page die je moet maken bevat slechts 1 element, *Grid*, dat is een tabel met rijen en kolommen. Dit element bevat 7 elementen.
* De eerste 3 elementen van de grid zijn definities op het niveau van de grid zelf, nl de rijen, de kolommen en de background.
  + Het element *Grid.RowDefinitions* definiëert het aantal rijen.
  + Zet hier 2 elementen met naam *RowDefinition* en attribuut *Height* met waarde 200. De grid heeft nu 2 rijen van 200 pixels.
  + Het element *Grid.ColumnDefinitions* definiëert het aantal kolommen.
  + Zet hier 2 elementen met naam *ColumnDefinition*. De grid heeft nu 2 kolommen die elk een halve page beslaan. Dit is typisch xaml, als je niets opgeeft van breedte of hoogte, beslist de engine zelf op de meest logische manier.
  + Het element *Grid.Background* definiëert de achtergrondkleur.
  + Zet hier een element *SolidColorBrush* en attribuut *Color* met waarde *ivory*.
* De 4 overblijvende elementen zijn de rechthoeken en ellipsen.
* Voor een rechthoek gebruik je het element *Rectangle* met attributen *Width*, *Height* en *Fill* voor de breedte (100), hoogte (30) en kleur (maroon en red).
* Voor een ellipse gebruik je het element *Ellipse* met dezelfde attributen als *Rectangle*.
* Om de rechthoeken en ellipsen te plaatsen in een cel, moet je werken met de attributen *Grid.Row* en *Grid.Column*. Een cel wordt geïndexeerd met het nummer van de rij en de kolom (zoals in Excel). Xaml begint te tellen vanaf 0.