

Eindverslag XML Receptenspeler



Receptenkaart

Toscaanse roodbaarsfilet met cheryltomaatjes, kapperjes en olijven

Aantal personen: 4 personen
Bereidingstijd: 20 minuten

Ingrediënten:

- 4 stuks roodbaarsfilet van 125 gram
- 1 el Bertoli Genile Extra Vergine olijfolie
- 1 bakje cheryltomaatjes
- 2 el bieslook, fijngehakt
- 2 el peterselle, fijngehakt
- 75 gram olijven
- 1 dosje zout
- 1 dosje versgemalen zwarte peper
- 2 el Bertoli Aceto Balsamico di Modena



Bereiding:

Verwarm de grill voor. Halveer de tomaatjes en de olijven. Doe ze vervolgens met de kapperjes, de balsamicoazijn, de bieslook en de peterselle in een kom en meng alles goed. Vat een tranché in met de olie, de azijn, de peper en de moedersaus. Verhaal

Tagliatelle met groene asperges, parmaham en groene kruiden

Aantal personen: 4 personen
Bereidingstijd: 20 minuten



Ingrediënten:

- 350 gram tagliatelle
- 700 gram groene asperges
- 300 gram parmaham, dun gesneden
- 300 ml slagroom
- 4 teelepels knoflook
- 2 el Bertoli Originale Extra Vergine olijfolie
- 3 el basilicum
- 3 el peterselle
- 150 gram verse parmezaanse kaas

Bereidingswijze:

Snij het onderste stukje van de asperges af, en snij ze daarna in stukken van 3 centimeter. Breng in een pannetje wat water aan de kook en Blancheer hierin de asperges gedurende 8 minuten.

Doe de slagroom in een pan. Voeg er drie hele teelepels knoflook aan toe en breng het geheel zachtjes aan de kook. Laat het heel zachtjes doorkoken tot de knoflook zacht is. Haal de pan van het vuur en haal de knoflook eruit (jeze heb je niet meer nodig).

Kook vervolgens de tagliatelle af deze volgens de gebruiksaanwijzing op de verpakking. Verwarm de extra vergine olijfolie in een pan en pers hierin het vierde teelepel knoflook uit. Fruit de knoflook even en voeg vervolgens de slagroom en de fijngehakte peterselle en basilicum toe.

Tip:

Bertoli Originale Extra Vergine olijfolie is ook erg goed te gebruiken bij bakken.

vrijg recept 4 / 5 vrijg recept

Vak: XML
Docent: Fons van Kesteren
Opdracht: Keuzeopdracht Receptenspeler
School: Hogeschool van Amsterdam
Opleiding: Interactieve Media

Datum: 23 oktober 2007
Versie: 1.0

Naam: Léon de Jong
Nummer: 503733

Florus Blankespoor
208480

Vincent Mei
505555

Inhoudsopgave

Projectplan	4
Debriefing	4
Rolverdeling	4
Gemaakte afspraken	4
Ontwerpdocumenten	5
Deelproducten en samenhang	5
Functioneel ontwerp	5
Technische keuzes.....	5
Taakverdeling	6
Planning	6
Gebruikte bronnen	7
Problemen en oplossingen	8
Suggesties voor verbeteringen	9
Code	10
Code XML bestand.....	10
Code ActionScript	12
Code XSLT	16

Projectplan

Debriefing

Wij hebben gekozen voor keuzeopdracht 3, de Receptenspeler. Bij deze opdracht is het de bedoeling dat je een 'ReceptenMarkupLanguage' ontwerpt. Deze presenteer je vervolgens in zowel een HTML pagina als een Flash applicatie. Deze zou je eventueel kunnen uitbreiden met zaken als een boodschappenlijst of kookwekker.

Rolverdeling

De rollen in ons team zijn als volgt:

Techniek Flash: Léon de Jong
Techniek XSLT: Floris Blankespoor
Design: Vincent Mei

Gemaakte afspraken

We hebben de volgende afspraken gemaakt:

- We werken met ActionScript 2.0 en XSLT
- We hebben samen de vocabulaire en de structuur van het XML document ontworpen
- We gebruiken Italiaanse recepten
- We gebruiken Italiaanse kleuren (rood, wit, groen)
- En natuurlijk de afspraken wie wat gaat doen

Ontwerpdocumenten

Deelproducten en samenhang

We hebben één XML bestand ontworpen voor kookrecepten en deze vervolgens gebruikt als bron voor een Flash applicatie en een 'HTML' document.

De Flash applicatie bestaat uit de volgende onderdelen:

- Het SWF bestand (stage + ActionScript)
- Het XML bestand
- De map 'images' met afbeeldingen

Het 'HTML' document bestaat uit de volgende onderdelen:

- Het XML bestand
- Het XSLT bestand
- Een CSS stylesheet
- De map 'images' met afbeeldingen

Functioneel ontwerp

We hebben een receptenviewer ontworpen in Flash. Hiermee kun je door Italiaanse kookrecepten bladeren. We hadden als eventuele extra mogelijkheden een boodschappenlijstje met automatische berekening van ingrediënten in gedachten (aantallen, hoeveelheden etc.) Hier zijn we helaas niet aan toe gekomen wegens tijdgebrek.

Het XSLT/HTML bestand toont een overzicht van de diverse recepten.

Technische keuzes

We hebben de volgende keuzes gemaakt tijdens de ontwerp van de applicatie

- Gebruik van Flash in combinatie met ActionScript 2.0
- Het gebruik van XPath in plaats van het standaard DOM van ActionScript 2.0
- Gebruik van XSLT voor het tonen van XML als een HTML pagina
- Gebruik van één en hetzelfde XML document voor beide applicaties

Taakverdeling

We hebben de opdracht als volgt verdeeld over het team:

- ActionScript: Léon de Jong
- XSLT & HTML: Florus Blankespoor
- Design: Vincent Mei
- Ontwerp XML: Team
- Eindverslag: Team
- Presentaties: Team

Planning

De planning was als volgt:

Week 6

Vorbereiden presentatie, bronnenonderzoek

Verantwoording: Team

Week 7

Presenteren. Eerste versie XML document af, elk teamlid gaat op zoek naar zijn eigen bronnen (documentatie, tutorials)

Verantwoording: Team

Week 8

Techniek van zowel de Flash als de XSLT applicatie moet af zijn, ontwerp XML document is definitief

Verantwoording: Léon en Floris

Week 9

Design van zowel de Flash applicatie als XSLT en CSS document is af. Presenteren eindresultaat.

Verantwoording design: Vincent

Verantwoording presentatie: Team

Gebruikte bronnen

We hebben de volgende bronnen gebruikt

Léon de Jong:

<http://www.w3schools.com/>

<http://www.kirupa.com/developer/flash/index.htm>

<http://www.bertolli.nl/> (voor de recepten)

Florus Blankespoor:

<http://protege.wordpress.com/category/xmltechniek/>

<http://www.w3schools.com/>

<http://www.bertolli.nl/> (voor de recepten)

Vincent Mei:

<http://www.flashkit.com/>

<http://www.bertolli.nl/> (voor de recepten)

Problemen en oplossingen

Probleem:

Bij het gebruik van het DOM van ActionScript 2.0 worden regels code erg lang, en in het geval van array's eventueel ingewikkeld.

Oplossing:

Gebruik van XPath in plaats van het standaard DOM van AS 2.0.

Probleem:

Het overgrote deel van de functies die XPath biedt, worden niet ondersteund.

Oplossing:

Hier is niet direct een oplossing voor, eventueel een plug-in voor een goede implementatie van XPath. Je zou eventueel gebruik kunnen maken van AS 3.0 met E4X implementatie. Deze implementatie maakt het gebruik van XML en Flash documenten stukken gemakkelijker.

Probleem:

Afbeeldingen met scherpe hoeken in een andere kleur dan die van de Flash applicatie.

Oplossing:

Het gebruik van een 'mask' of 'graphic overlay' om deze te verbergen.

Probleem:

Witruimtes die meegenomen worden in het Flashdocument, bij het inspringen van inhoud van tags in een XML document.

Oplossing:

In het XML bestand tags direct voor en achter de inhoud plaatsen.

Suggesties voor verbeteringen

Omdat we het nogal druk hadden met andere vakken (zes oplevermomenten voor zes vakken in één week) hebben we geen tijd kunnen vrijmaken om onze Flashapplicatie uit te breiden met bijvoorbeeld een kookwekker of boodschappenlijstje met automatische berekening van ingrediënten. Als we eerder hadden kunnen beginnen met de opdracht, bijvoorbeeld in week drie, hadden we een uitgebreidere applicatie kunnen afleveren.

Wanneer we gebruik hadden gemaakt van ActionScript 3.0 hadden we nettere code kunnen afleveren. Maar dat ligt dus aan ActionScript en niet aan ons. We hebben de code zo compact en clean mogelijk gehouden als mogelijk was.

Code

Code XML bestand

Onderstaande code is niet het volledige XML bestand, het volledige bestand is vijf keer zo lang. Hieronder zie je de code van één recept.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="receptenstylesheet.xslt"?>

<recepten>
  <recept>
    <gerecht>Toscaanse roodbaarsfilet met cherrytomaatjes, kappertjes en olijven</gerecht>
    <afbeelding>images/1.jpg</afbeelding>
    <personen>4 personen</personen>
    <tijd>20 minuten</tijd>

    <ingredienten>
      <ingredient>
        <aantal>4</aantal>
        <eenheid>stuks</eenheid>
        <naam>roodbaarsfilets van 125 gram</naam>
      </ingredient>
      <ingredient>
        <aantal>1</aantal>
        <eenheid>el</eenheid>
        <naam>Bertolli Gentile Extra Vergine olijfolie</naam>
      </ingredient>
      <ingredient>
        <aantal>1</aantal>
        <eenheid>bakje</eenheid>
        <naam>cherrytomaatjes</naam>
      </ingredient>
      <ingredient>
        <aantal>2</aantal>
        <eenheid>el</eenheid>
        <naam>bieslook, fijnggehakt </naam>
      </ingredient>
      <ingredient>
    
```

```
<aantal>2</aantal>
<eenheid>el</eenheid>
<naam>peterselie, fijngehakt</naam>

</ingredient>

<ingredient>

    <aantal>75</aantal>
    <eenheid>gram</eenheid>

    <naam>olijven</naam>

</ingredient>

<ingredient>

    <aantal>1</aantal>
    <eenheid>dosis</eenheid>
    <naam>zout</naam>

</ingredient>

<ingredient>

    <aantal>1</aantal>
    <eenheid>dosis</eenheid>
    <naam>versgemalen zwarte peper</naam>

</ingredient>

<ingredient>

    <aantal>2</aantal>
    <eenheid>el</eenheid>
    <naam>Bertolli Aceto Balsamico di Modena</naam>

</ingredient>

</ingredienten>

<bereiding>

    <p>
    Verwarm de grill voor. Halveer de tomaatjes en de olijven. Doe ze vervolgens met de kappertjes,
    de balsamicoazijn, de bieslook en de peterselie in een kom en meng alles goed.
    </p>

    <p>
    Vet een bakblik in met Gentile extra vergine olijfolie en leg de roodbaarsfilets erop. Verdeel het
    tomatenmengsel over de visfilets. Schuif het bakblik 10 cm onder de hete grill en rooster de vis in
    8-10 minuten mooi bruin en net gaar. Lekker met Roseval aardappeltjes.
    </p>

</bereiding>

<tip>
Bertolli Gentile extra vergine olijfolie is zachter en milder van smaak en past perfect bij gerechten met vis.
</tip>

</recept>

</recepten>
```

Code ActionScript

```
import mx.xpath.XPathAPI;

function loadXML(loaded) {

    if (loaded) {

        xmlNode = this.firstChild;
        image = [];
        description = [];
        persons = [];
        time = [];
        ingredients = [];
        tips = [];
        bread = [];
        count = [];
        unit = [];
        name = [];
        total = xmlNode.childNodes.length;

        for (i=0; i<total; i++) {

            image[i] = mx.xpath.XPathAPI.selectSingleNode(xmlNode.childNodes[i],
                "recept/afbeelding").firstChild.nodeValue;

            description[i] = mx.xpath.XPathAPI.selectSingleNode(xmlNode.childNodes[i],
                "recept/gerecht").firstChild.nodeValue;

            persons[i] = mx.xpath.XPathAPI.selectSingleNode(xmlNode.childNodes[i],
                "recept/personen").firstChild.nodeValue;

            time[i] = mx.xpath.XPathAPI.selectSingleNode(xmlNode.childNodes[i],
                "recept/tijd").firstChild.nodeValue;

            tips[i] = mx.xpath.XPathAPI.selectSingleNode(xmlNode.childNodes[i],
                "recept/tip").firstChild.nodeValue;

            count[i] = mx.xpath.XPathAPI.selectNodeList(xmlNode.childNodes[i],
                "recept/ingredienten/*/aantal/*");

            unit[i] = mx.xpath.XPathAPI.selectNodeList(xmlNode.childNodes[i],
                "recept/ingredienten/*/eenheid/*");

            name[i] = mx.xpath.XPathAPI.selectNodeList(xmlNode.childNodes[i],
                "recept/ingredienten/*/naam/*");

            bread[i] = mx.xpath.XPathAPI.selectNodeList(xmlNode.childNodes[i],
                "recept/bereiding/p/*");

        }

        firstImage();

    } else {

        content = "file not loaded!";

    }

}
```

```
xmlData = new XML();

xmlData.ignoreWhite = true;
xmlData.onLoad = loadXML;
xmlData.load("recepten.xml");

listen = new Object();
listen.onKeyDown = function() {

    if (Key.getCode() == Key.LEFT) {

        prevImage();

    } else if (Key.getCode() == Key.RIGHT) {

        nextImage();

    }

};

Key.addListener(listen);
prev_btn.onRelease = function() {

    prevImage();

};

next_btn.onRelease = function() {

    nextImage();

};

p = 0;
this.onEnterFrame = function() {

    filesize = picture.getBytesTotal();
    loaded = picture.getBytesLoaded();
    preloader._visible = true;

    if (loaded != filesize) {

        preloader.preload_bar._xscale = 100*loaded/filesize;

    } else {

        preloader._visible = false;

        if (picture._alpha < 100) {

            picture._alpha += 10;

        }

    }

};

function nextImage() {

    if (p < (total-1)) {
```

```
p++;
bread_txt.text = [];
ingredients_txt.text = [];

if (loaded == filesize) {

    picture._alpha = 0;
    picture.loadMovie(image[p],1);
    desc_txt.text = description[p];
    persons_txt.text = persons[p];
    time_txt.text = time[p];
    tips_txt.text = tips[p];

    for (t=0; t<xmlNode.childNodes[p].childNodes[4].childNodes.length; t++) {

        ingredients_txt.text += count[p][t]+" "+unit[p][t]+" "+name[p][t)+"\n";

    }

    for (t=0; t<bread[p].length; t++) {

        bread_txt.text += bread[p][t)+"\n\n";

    }

    picture_num();

}

}

function prevImage() {

    if (p>0) {

        p--;
        bread_txt.text = [];
        ingredients_txt.text = [];
        picture._alpha = 0;
        picture.loadMovie(image[p],1);
        desc_txt.text = description[p];
        persons_txt.text = persons[p];
        time_txt.text = time[p];
        tips_txt.text = tips[p];

        for (t=0; t<xmlNode.childNodes[p].childNodes[4].childNodes.length; t++) {

            ingredients_txt.text += count[p][t]+" "+unit[p][t]+" "+name[p][t)+"\n";

        }

        for (t=0; t<bread[p].length; t++) {

            bread_txt.text += bread[p][t)+"\n\n";

        }

        picture_num();

    }

}
```

```
function firstImage() {
    if (loaded == filesize) {
        picture._alpha = 0;
        picture.loadMovie(image[0],1);
        desc_txt.text = description[0];
        persons_txt.text = persons[0];
        time_txt.text = time[p];
        tips_txt.text = tips[p];

        for (t=0; t<xmlNode.childNodes[0].childNodes[4].childNodes.length; t++) {
            ingredients_txt.text += count[0][t]+" "+unit[0][t]+" "+name[0][t)+"\n";
        }

        for (t=0; t<bread[0].length; t++) {
            bread_txt.text += bread[0][t)+"\n\n";
        }

        picture_num();
    }
}

function picture_num() {
    current_pos = p+1;
    pos_txt.text = current_pos+" / "+total;
}
```

Code XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">

  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">

    <html>

      <head>

        <title>Bertolli</title>
        <link rel="stylesheet" href="css/default.css" type="text/css"/>

      </head>

      <body>

        <div id="wrapper">

          <div id="header">

            

          </div>

          <div id="screen">

            <div id="back">

              <h1>Receptenkaart</h1>
              <xsl:apply-templates />

            </div>

          </div>

          <div id="footer"></div>

        </div>

      </body>

    </html>

  </xsl:template>

  <xsl:template match="recept">

    <hr/>
    <h2><xsl:apply-templates select="gerecht"/></h2>
    <em>Aantal personen: </em><xsl:apply-templates select="personen"/><br/>
    <em>Bereidingstijd: </em><xsl:apply-templates select="tijd"/><br/><br/>
    <img id="gerecht">

      <xsl:attribute name="src">
        <xsl:value-of select="afbeelding"/>
      </xsl:attribute>

    </img>

  </xsl:template>

</xsl:stylesheet>
```

```
<xsl:apply-templates select="ingredienten"/>

<xsl:apply-templates select="bereiding"/>
<xsl:apply-templates select="tip"/><br/><br/>

</xsl:template>

<xsl:template match="ingredienten">

    <h3>Ingredienten:</h3>
    <xsl:apply-templates select="ingredient"/>

</xsl:template>

<xsl:template match="ingredient">

<ul type="square">

    <li>
        <xsl:apply-templates select="aantal"/><xsl:text> </xsl:text><xsl:apply-templates
        select="eenheid"/><xsl:text> </xsl:text><xsl:apply-templates select="naam"/>
    </li>

</ul>

</xsl:template>

<xsl:template match="bereiding">

    <h3>Bereiding:</h3>
    <xsl:apply-templates select="p"/>
    <h3>Tip</h3>
    <xsl:apply-templates select="tip"/>

</xsl:template>

<xsl:template match="p">

    <p><xsl:apply-templates /></p>

</xsl:template>

</xsl:stylesheet>
```