

Generación de interfaces a partir de documentos XML

DESARROLLO DE APLICACIONES MULTIPLATAFORMA



ÍNDICE

- Introducción
- XML
 - Ventajas
 - Estructura
 - DOM
- Lenguajes de descripción de interfaces basados en XML
 - XAML
 - XUL
 - SVG
 - OTROS: UIML, MXML, FXML

INTRODUCCIÓN

OBJETIVOS DE LA UNIDAD DIDÁCTICA

- Reconocer las ventajas de generar interfaces gráficas de usuario mediante editores visuales
- Capacidad de generar la descripción de la interfaz en XML usando un editor gráfico
- Capacidad de analizar y modificar el documento XML generado
- Capacidad de asignar acciones a los eventos de la interfaz
- Capacidad de generar el código correspondiente a la interfaz a partir del documento XML
- Capacidad de programar una aplicación que incluye la interfaz generada

XML eXtensible Markup Language

- Lenguaje de marcas desarrollado por el *World Wide Web* (W3C)
- Al igual que HTML deriva del SGML (Standarda Generalized Markup Language)
- Estándar para el intercambio de información estructurada entre diferentes plataformas
- Aplicación no solo para Internet, entre otros en bases de datos, editores de texto y hojas de cálculos
- Permite compatibilidad entre sistemas para compartir información de manera segura, fiable y ágilmente

XML VENTAJAS

- Diseñado para ser entendible tanto por humanos como software
- Simplifica el compartir e intercambiar información
- Separa claramente el contenido de su presentación o formato
- Es extensible
- Analizador estándar
- Desde 2011 recomendación de [W3C](#)
- Mejora la compatibilidad entre aplicaciones

XML ESTRUCTURA

- Un documento XML tiene dos estructuras, una lógica y otra física
- Físicamente, un documento XML puede consistir en una o mas unidades de almacenamiento llamadas **entidades**
- Las entidades tienen contenido y están identificadas por un nombre
- Todo documento XML tiene una entidad principal, llamada entidad documento, que sirve como punto de partida para el procesador XML
- Lógicamente, un documento XML está estructurado en forma de árbol
- El documento XML está compuesto por declaraciones, elementos, comentarios, referencias e instrucciones de procesamiento

XML ESTRUCTURA

- Los elementos pueden tener atributos
- Cada documento XML contiene uno o mas elementos, cuyos límites están delimitados por etiquetas de comienzo y final
- Cada elemento tiene un tipo, identificado por un nombre, denominado identificador genérico, y puede tener un conjunto de especificaciones de atributos
- Cada documento XML tiene asociado una “declaración de tipo de documento”, dentro de un documento DTD (Document Type Definition), que define qué tipo de documento es

XML ESTRUCTURA - EJEMPLO

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Msg SYSTEM "Edit_Msg.dtd">
<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Vincent Vega</Nombre>
      <Mail>vv@pulpfiction.com</Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Jules Winnfield</Nombre>
      <Mail>jw@pulpfiction.com</Mail>
    </Destinatario>
    <Texto>
      <Asunto>Cena con Mia Wallace</Asunto>
      <Parrafo>Mañana tengo que...</Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

Edit_Msg.xml

```
<?xml version="1.0" encoding="ISO-
8859-1" ?>
<!-- Este es el DTD de Edit_Msg -->
<!ELEMENT Mensaje (Remitente,
Destinatario, Texto)*>
<!ELEMENT Remitente (Nombre, Mail)>
<!ELEMENT Remitente (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Destinatario (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Texto (Asunto, Parrafo)>
<!ELEMENT Asunto (#PCDATA)>
<!ELEMENT Parrafo (#PCDATA)>
```

Edit_Msg.dtd

XML DOM (DOCUMENT OBJECT MODEL)

- DOM es una interfaz de programación para documentos HTML y XML
- Proporciona una representación estructurada del documento, HTML o XML, y define la manera en que esa estructura puede ser accedida por diferentes programas para poder cambiar la estructura, estilo y contenido del documento
- La representación del documento de DOM se compone de un grupo estructurado de nodos y objetos con sus propiedades y métodos
- La mayoría de los navegadores actuales implementan DOM mediante los estándares [W3C DOM](#) y [WHATWG DOM](#)

LENGUAJES DE DESCRIPCIÓN BASADOS EN XML

VENTAJAS

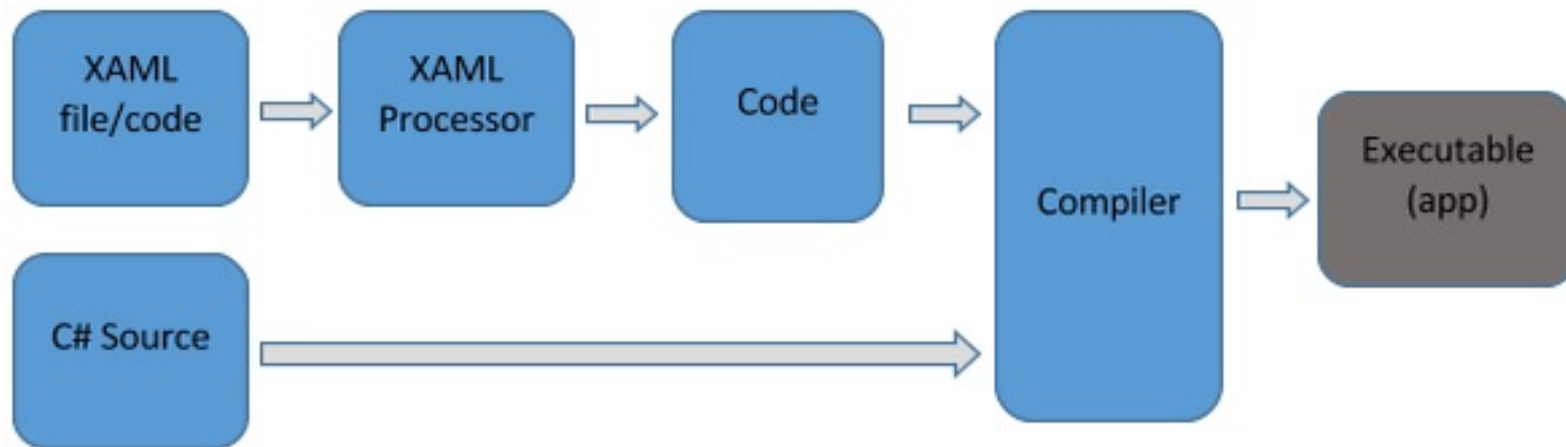
- Son de fácil aprendizaje
- Permiten definir la interfaz de forma independiente de la lógica y contenido de la aplicación
- XAML
- XUL
- UIML
- SVG
- MXML

LENGUAJES: XAML

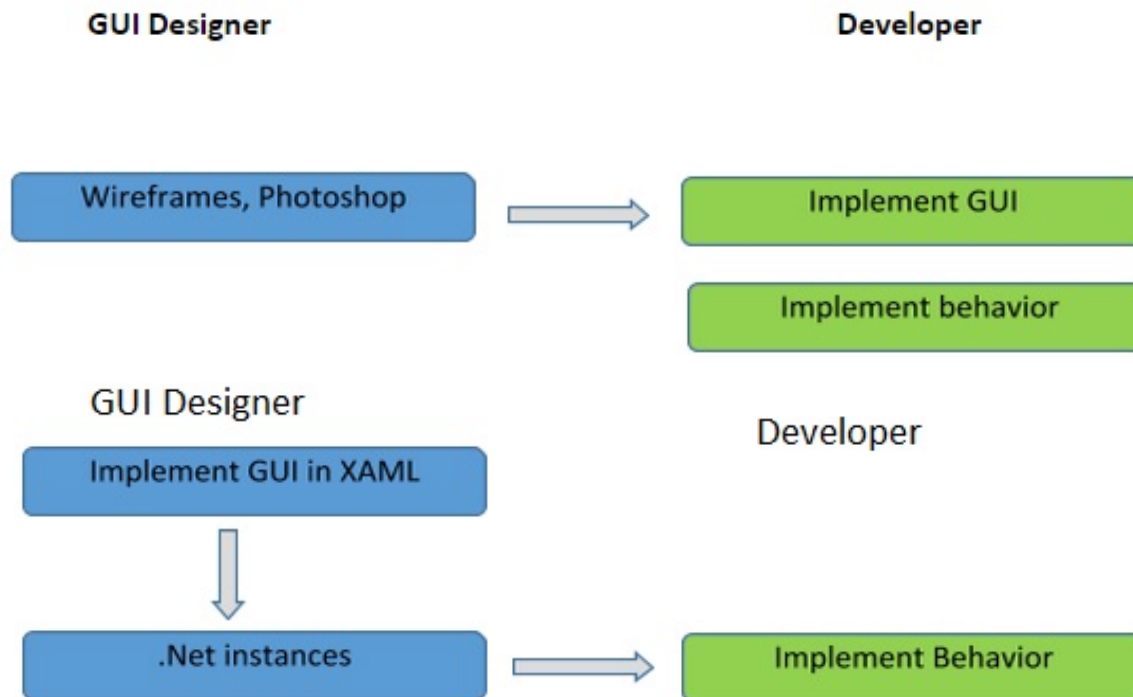
XTENSIBLE APPLICATION MARKUP LANGUAGE

- Lenguaje para la interfaz de usuario de Windows Presentation Foundation (WPF)
- Lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuario
- Diseñado para soportar las clases y métodos de la plataforma de desarrollo .NET
- Independiente de la lógica de la aplicación
- Para aplicaciones de escritorio y web

LENGUAJES: XAML INFRAESTRUCTURA



LENGUAJES: XAML VENTAJAS

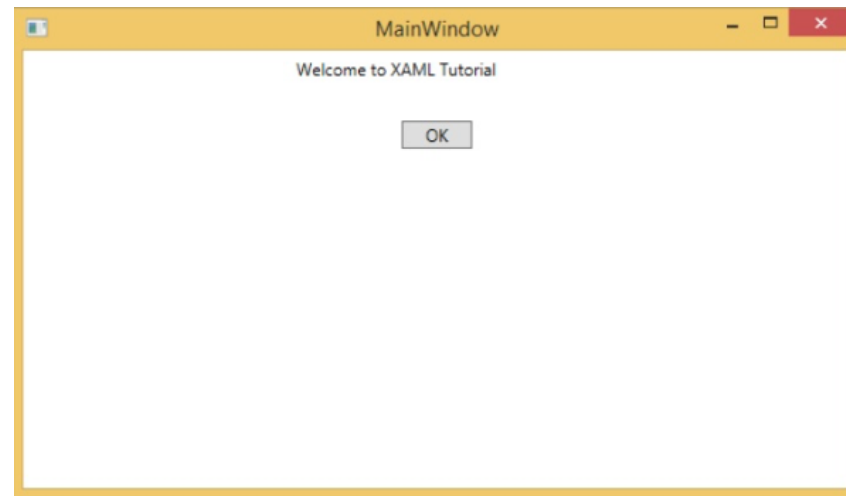


LENGUAJES: XAML EJEMPLO

```
<Window x:Class = "XAMLVsCode.MainWindow"
  xmlns = "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x = "http://schemas.microsoft.com/winfx/2006/xaml" Title = "MainWindow" Height = "350" Width = "525">

  <StackPanel>
    <TextBlock Text = "Welcome to XAML Tutorial" Height = "20" Width = "200" Margin = "5"/>
    <Button Content = "Ok" Height = "20" Width = "60" Margin = "5"/>
  </StackPanel>

</Window>
```



LENGUAJES: XAML VS VISUAL BASIC.NET

```
Public Class MainWindow

    Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
        Dim panel As New StackPanel()
        panel.Orientation = Orientation.Vertical
        Me.Content = panel
        Dim txtInput As New TextBlock

        txtInput.Text = "Welcome to XAML Tutorial with VB.net"
        txtInput.Width = 220
        txtInput.Height = 20
        txtInput.Margin = New Thickness(5)

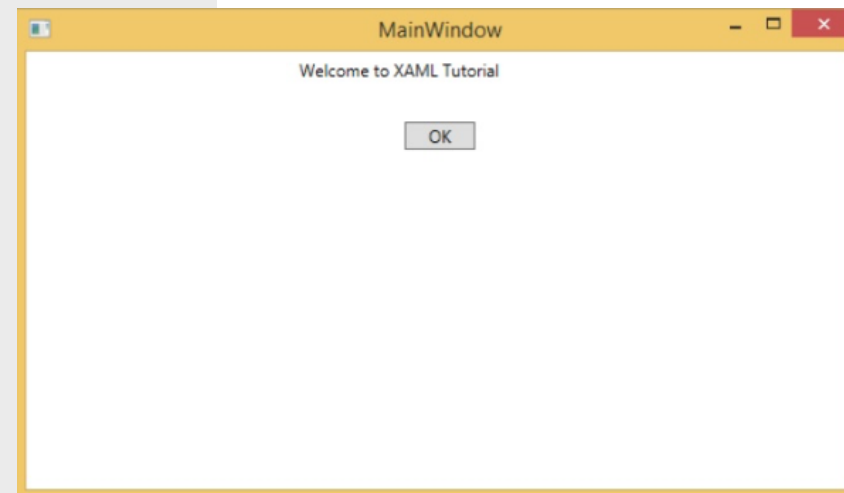
        panel.Children.Add(txtInput)
        Dim btn As New Button()

        btn.Content = "Ok"
        btn.Width = 60
        btn.Height = 20
        btn.Margin = New Thickness(5)

        panel.Children.Add(btn)

    End Sub

End Class
```



LENGUAJES: XAML BUILDING BLOCKS

- CONTENEDORES

https://www.tutorialspoint.com/xaml/xaml_layouts.htm

- STACKPANEL
- WRAPPANEL
- DOCKPANEL
- CANVASPANEL
- GRIDPANEL

- ELEMENTOS UI / CONTROLES */xaml/xaml_controls.htm

- BUTTON
- CALENDAR
- CHECKBOX
- ...

LENGUAJES: XAML BUILDING BLOCKS

- EVENTOS */xaml/xaml_event_handling.htm
 - CLICK
 - MOUSEDOWN
 - MOUSEENTER
 - MOUSELEAVE
 - MOUSEUP
 - KEYDOWN
 - KEYUP

LENGUAJES: XAML OTRAS FUNCIONALIDADES

- DATA BINDING */xaml/xaml_data_binding.htm
- PLANTILLAS */xaml/xaml_templates.htm
- ESTILOS */xaml/xaml_styles.htm
- TRIGGERS */xaml/xaml_triggers.htm
- DEBUGGING */xaml/xaml_debugging.htm
- CONTROLES PERSONALIZADOS
*/xaml/xaml_custom_controls.htm

LENGUAJES: XUL

XML-BASED USER-INTERFACE LANGUAGE

- Lenguaje para la interfaz de usuario en el navegador Mozilla
- No es un estándar
- Aporta una definición de interfaces GUI simple y portable
- Se puede utilizar en otras aplicaciones utilizando JavaScript
- A diferencia de HTML, provee un gran conjunto de herramientas para crear menús, paneles, barras de herramientas y wizards
- El propio navegador Firefox o el cliente de correo Thunderbird han sido desarrollados en XUL

LENGUAJES: XUL ESTRUCTURA

- El navegador Mozilla no es mas un conjunto de paquetes coteniendo ficheros XUL, JavaScript y CSS
- Mozilla usa el sistema de ficheros [Chrome](#) que permite acceder a URL especiales: *chrome://URL*
- Mozilla renderiza tanto HTML como XUL usando el mismo motor y utilizando CSS para especificar la presentación
- El sistema Chrome también puede ser utilizado para acceder a paquetes ya instalados y abrirlos

LENGUAJES: XUL ORGANIZACIÓN PAQUETES

Todo parte del directorio Chrome:

- CONTENT – Ventanas y scripts

```
1 content
2   browser
3     browser.xul
4     browser.js
5     -- other browser XUL and JS files goes here --
6   bookmarks
7     -- bookmarks files go here --
8   preferences
9     -- preferences files go here --
10 .
```

- SKIN – Hojas de estilo, imágenes y ficheros de otros temas

```
1 skin
2   classic
3     browser
4       browser.css
5       -- other browser skin files go here --
6   global
7     -- global skin files go here --
8 .
```

- LOCATE – Documentos DTD y archivos de localización

LENGUAJES: XUL EJEMPLO

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
3 <window id="findfile-window"
4       title="Find Files"
5       orient="horizontal"
6       xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
7
8   <button id="find-button" label="Find"/>
9   <button id="cancel-button" label="Cancel"/>
10
11 </window>
```



LENGUAJES: XUL ELEMENTOS SIMPLES

- Botones https://developer.mozilla.org/en-US/docs/XUL/Tutorial/Adding_Buttons
- Imágenes [*/Tutorial/Adding_Labels_and_Images](https://developer.mozilla.org/en-US/docs/XUL/Tutorial/Adding_Labels_and_Images)
- Controles de entrada [*/Tutorial/Input_Controls](https://developer.mozilla.org/en-US/docs/XUL/Tutorial/Input_Controls)

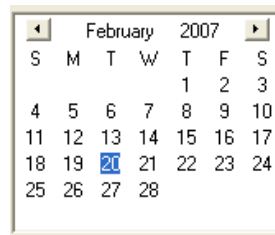
```
1 <label value="Search for:" control="find-text"/>
2 <textbox id="find-text"/>
3
4 <button id="find-button" label="Find"/>
```



LENGUAJES: XUL ELEMENTOS SIMPLES

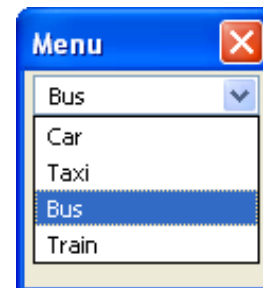
- Controles numéricos */Tutorial/Numeric_Controls

```
1 | <datepicker type="popup"/>
```



- Controles de listas */Tutorial/List_Controls

```
1 | <menulist label="Bus">
2 |   <menupopup>
3 |     <menuitem label="Car" />
4 |     <menuitem label="Taxi" />
5 |     <menuitem label="Bus" selected="true" />
6 |     <menuitem label="Train" />
7 |   </menupopup>
8 | </menulist>
```



LENGUAJES: XUL ELEMENTOS SIMPLES

- Medidores de progreso [*/Tutorial/Progress Meters](#)



- Añadir html a una ventana [*/Tutorial/Adding_HTML_Elements](#)



LENGUAJES: XUL ELEMENTOS SIMPLES

- Espaciadores [*/Tutorial/Using Spacers](#)
- Más sobre botones [*/Tutorial/More Button Features](#)

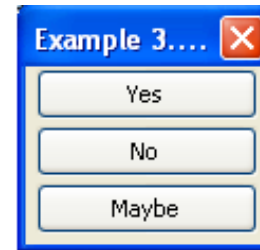
```
1 <button id="find-button"  
2   label="Find" style="list-style-image: url('happy.png')"/>
```



LENGUAJES: XUL OTROS ELEMENTOS

- Modelo de caja [*/Tutorial/The_Box_Model](#)

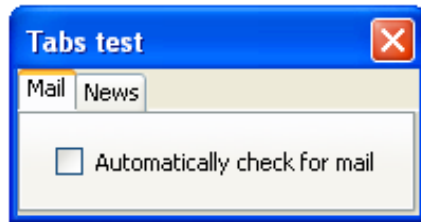
```
1 <vbox>
2   <button id="yes" label="Yes"/>
3   <button id="no" label="No"/>
4   <button id="maybe" label="Maybe"/>
5 </vbox>
```



- Posicionamiento de los elementos [*/Tutorial/Element_Positioning](#)
- Detalles del modelo de caja [*/Tutorial/Box_Model_Details](#)
- Cajas de grupo [*/Tutorial/Groupboxes](#)
- Añadiendo más elementos [*/Tutorial/Adding_More_Elements](#)

LENGUAJES: XUL ELEMENTOS DE LAYOUT

- Stacks y Decks [*/Tutorial/Stacks_and_Decks](#)
- Posicionamiento del Stack [*/Tutorial/Stack_Positioning](#)
- Contenedores de pestañas [*/Tutorial/Tabboxes](#)

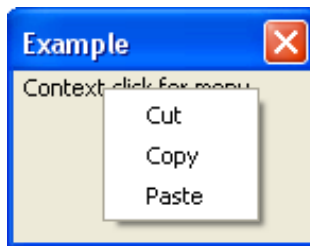


```
1 <tabbox>
2   <tabs>
3     <tab label="Mail"/>
4     <tab label="News"/>
5   </tabs>
6   <tabpanel>
7     <tabpanel id="mailtab">
8       <checkbox label="Automatically check for mail"/>
9     </tabpanel>
10    <tabpanel id="newstab">
11      <button label="Clear News Buffer"/>
12    </tabpanel>
13  </tabpanel>
14 </tabbox>
```

- Cuadrículas [*/Tutorial/Grids](#)
- Paneles de contenidos [*/Tutorial/Content_Panels](#)
- Divisores [*/Tutorial/Splitters](#)

LENGUAJES: XUL TOOLBARS Y MENUS


- Barra de herramientas [*/Tutorial/Toolbars](#)
- Menús de barras simples [*/Tutorial/Simple Menu Bars](#)
- Más funciones de menús [*/Tutorial/More Menu Features](#)
- Menús emergentes [*/Tutorial/Popup_Menus](#)



```
1 <popupset>
2   <menupopup id="clipmenu">
3     <menuitem label="Cut"/>
4     <menuitem label="Copy"/>
5     <menuitem label="Paste"/>
6   </menupopup>
7 </popupset>
8
9 <box context="clipmenu">
10   <label value="Context click for menu"/>
11 </box>
```

- Menús con desplazamiento [*/Tutorial/Scrolling_Menus](#)

LENGUAJES: XUL EVENTOS

- Añadiendo eventos */Tutorial/Adding_Event_Handlers
- Se necesita la utilización de scripts, en concreto JavaScript
- Se incorpora el elemento script 
- El script incorpora el código que responde a los mas de treinta eventos que hay definidos entre los diferentes elementos XUL
- Ejemplo de evento: 'mousemove'
- Tres fases:
 - CAPTURING PHASE
 - TARGET PHASE
 - BUBBLING PHASE

LENGUAJES: XUL EVENTOS LISTENERS

- ATRIBUTOS EVENT LISTENERS

```
1 <menuitem label="Close" accesskey="c" oncommand="window.close();"/>
2 ...
3
4 <button id="cancel-button" label="Cancel"
5         oncommand="window.close();"/>
```

- DOM EVENT LISTENERS

```
1 <button id="okbutton" label="OK"/>
2
3 <script>
4 function buttonPressed(event){
5     alert('Button was pressed!');
6 }
7
8 var button = document.getElementById("okbutton");
9 button.addEventListener('command', buttonPressed, true);
10 </script>
```

LENGUAJES: XUL EVENTOS

- Mas eventos [*/Tutorial/More_Event_Handlers](#)
 - click, dbclick, mousedown, mouseup, mouseover,
 - mousemove, mouseout
- Atajos del teclado [*/Tutorial/Keyboard_Shortcuts](#)
- Selección y Focus [*/Tutorial/Focus_and_Selection](#)
- Elementos Commands [*/Tutorial/Commands](#)
- Modificando Commands [*/Tutorial/Updating_Commands](#)
- Broadcasters y Observers [*/Tutorial/Broadcasters_and_Observers](#)

Event	On Event Handler
drag	ondrag
dragend	ondragend
dragenter	ondragenter
dragexit	ondragexit
dragleave	ondragleave
dragover	ondragover
dragstart	ondragstart
drop	ondrop

LENGUAJES: XUL DOM

- DOM [*/Tutorial/Document_Object_Model](#)

`getAttribute (name)`

Return the value of the attribute with the given name.

`hasAttribute (name)`

Return true if the attribute with the given name has a value.

`setAttribute (name , value)`

Set the value of the attribute with the given name to the given value.

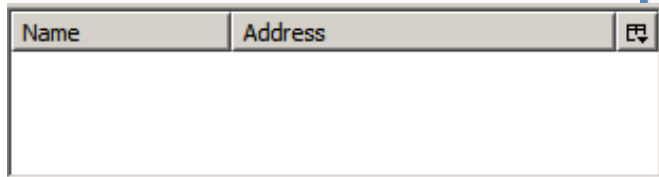
`removeAttribute (name)`

Remove the attribute with the given name.

- Modificando la interfaz XUL [*/Tutorial/Modifying_a_XUL_Interface](#)
- Manipulando listas [*/Tutorial/Manipulating_Lists](#)
- Objetos Caja [*/Tutorial/Box_Objects](#)
- Interfaces XPCOM [*/Tutorial/XPCOM_Interfaces](#)
- Ejemplos XPCOM [*/Tutorial/XPCOM_Examples](#)

LENGUAJES: XUL ÁRBOLES

- Árboles [*/Tutorial/Trees](#)



```
1 <tree flex="1">
2
3   <treecols>
4     <treecol id="nameColumn" label="Name" flex="1"/>
5     <treecol id="addressColumn" label="Address" flex="2"/>
6   </treecols>
7
8   <treechildren/>
9
10 </tree>
```

- Características de los árboles [*/Tutorial/More_Tree_Features](#)
- Selección en un árbol [*/Tutorial/Tree_Selection](#)
- Vistas personalizadas en un árbol [*/Tutorial/Custom_Tree_Views](#)
- Detalles de vistas en un árbol [*/Tutorial/Tree_View_Details](#)
- Objetos Caja en un árbol [*/Tutorial/Tree_Box_Objects](#)

LENGUAJES: XUL ASPECTOS Y TEMAS

- Añadiendo hojas de estilos [*/Tutorial/Adding_Style_Sheets](#)
- Estilizando un árbol [*/Tutorial/Styling_a_Tree](#)
- Modificando el tema por defecto
[*/Tutorial/Modifying_the_Default_Skin](#)
- Creando un tema [*/Tutorial/Creating_a_Skin](#)
- Localización [*/Tutorial/Localization](#)
- Ficheros propietarios [*/Tutorial/Property_Files](#)

LENGUAJES: XUL OTRAS PROPIEDADES

- Plantillas y RDF
- Vinculaciones
- Tipos de ventanas especializadas
- Instalación
- https://developer.mozilla.org/es/docs/Tutorial_de_XUL

LENGUAJES: XUL APLICACIONES

Aplicaciones mas comunes de XUL :

- **EXTENSIONES DE FIREFOX:** elementos como barras de herramientas o menús que agregan diferentes funcionalidades a Firefox
- **APLICACIONES INDEPENDIENTES:** componentes como XULRunner permiten crear aplicaciones XUL independientes
- **PAQUETE XUL:** instalado como una extensión, es un enfoque intermedio entre las dos aplicaciones anteriores, que actúa como una aplicación separada del navegador
- **APLICACIONES XUL REMOTAS:** aplicaciones alojadas en un servidor web y ejecutables remotamente

LENGUAJES: XUL AVANZADO

AÑADIR UNA NUEVA FUNCIONALIDAD AL NAVEGADOR:

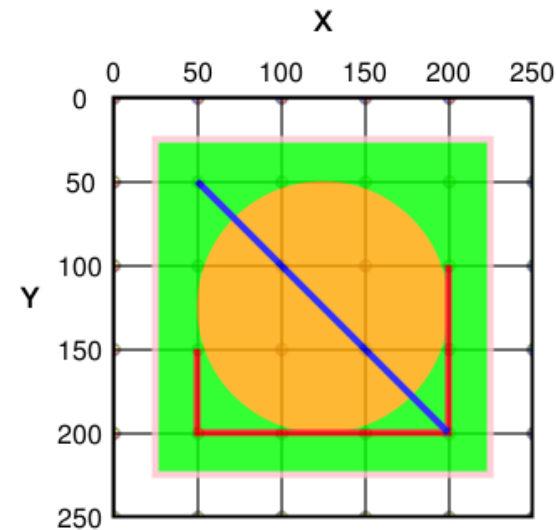
- SISTEMA CHROME: <https://developer.mozilla.org/en/Chrome>
- XUL OVERLAY: https://developer.mozilla.org/en/XUL_Overlays
- DESARROLLO DE ADD-ONS: a partir de 2017 WebExtensions será el estándar para el desarrollo de add-ons para Firefox y se dejará de lado el sistema de XUL Overlay utilizado hasta ahora.
- WEBEXTENSIONS: <https://developer.mozilla.org/en-US/Add-ons/WebExtensions>

LENGUAJES: SVG

SCALABLE VECTOR GRAPHICS

- Especificación basada en XML para describir gráficos vectoriales bidimensionales, tanto estáticos como animados
- Lenguaje vectorial para definir figuras simples especificadas mediante ecuaciones matemáticas o expresiones algebraicas
- Permite crear figuras complejas a partir de un conjunto de figuras simples (cuadrados, rectángulos, triángulos y círculos entre otros)
- Soportado por la mayoría de navegadores sin necesidad de complementos externos
- Desde 2011 recomendación de [W3C](#)

LENGUAJES: SVG EJEMPLOS



```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />
  <circle cx="125" cy="125" r="75" fill="orange" />
  <polyline points="50,150 50,200 200,200 200,100" stroke="red" stroke-width="4" fill="none" />
  <line x1="50" y1="50" x2="200" y2="200" stroke="blue" stroke-width="4" />
</svg>
```


LENGUAJES: SVG ELEMENTOS SIMPLES

- Rectángulo <rect>



```
<svg width="400" height="180">  
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"  
    style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
</svg>
```

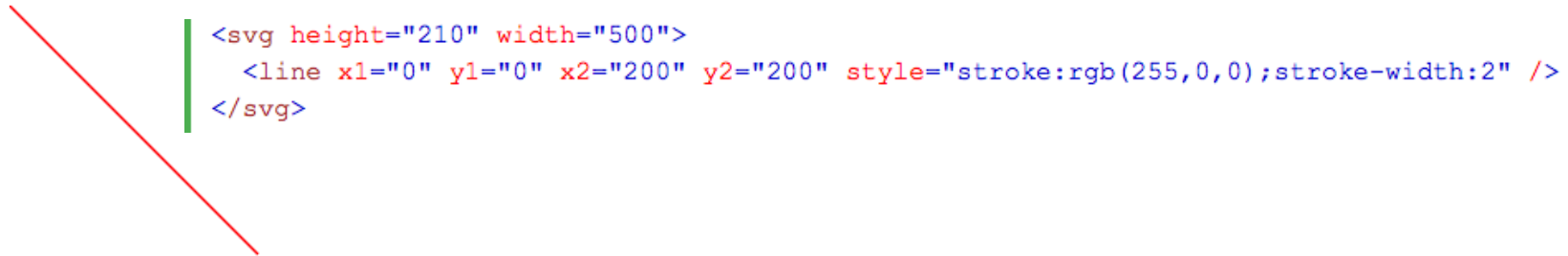
- Círculo <circle>



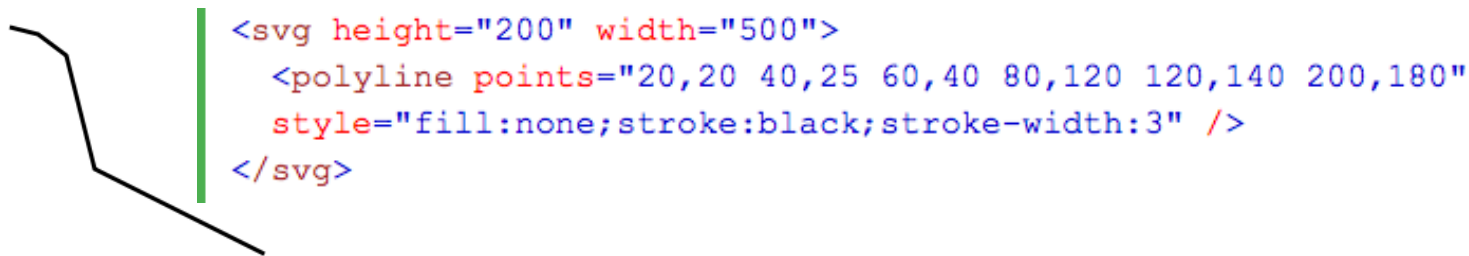
```
<svg height="100" width="100">  
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />  
</svg>
```

LENGUAJES: SVG ELEMENTOS SIMPLES

- Elipse <ellipse>
- Línea <line>

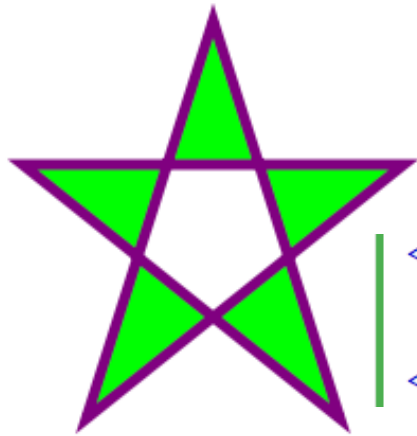


- Polilínea <polyline>



LENGUAJES: SVG ELEMENTOS SIMPLES

- Polígono <polygon>



```
<svg height="210" width="500">  
  <polygon points="100,10 40,198 190,78 10,78 160,198"  
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />  
</svg>
```

- Path <path>

LENGUAJES: SVG OTROS ELEMENTOS

- Texto <text>

I love SVG

```
<svg height="60" width="200">  
  <text x="0" y="15" fill="red" transform="rotate(30 20,40)">I love SVG</text>  
</svg>
```

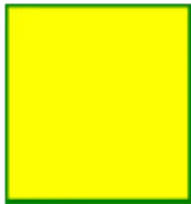
- Trazo <path stroke> (propiedad)

- stroke
- stroke-width
- stroke-linecap
- stroke-dasharray

```
<svg height="80" width="300">  
  <g fill="none" stroke="black" stroke-width="4">  
    <path stroke-dasharray="5,5" d="M5 20 1215 0" />  
    <path stroke-dasharray="10,10" d="M5 40 1215 0" />  
    <path stroke-dasharray="20,10,5,5,5,10" d="M5 60 1215 0" />  
  </g>  
</svg>
```

LENGUAJES: SVG OTROS ELEMENTOS

- Filtros
- Efectos de desenfoque
- Sombras paralelas



```
<svg height="140" width="140">
  <defs>
    <filter id="f2" x="0" y="0" width="200%" height="200%">
      <feOffset result="offOut" in="SourceGraphic" dx="20" dy="20" />
      <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10" />
      <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3"
    fill="yellow" filter="url(#f2)" />
</svg>
```

- Lineal
- Radial

LENGUAJES: SVG EMBEBIDO EN HTML5



```
<!DOCTYPE html>
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

LENGUAJES: SVG REFENCIAS

- TODOS LOS ELEMENTOS
- http://www.w3schools.com/graphics/svg_reference.asp
- MÁS EJEMPLOS
- http://www.w3schools.com/graphics/svg_examples.asp
- TUTORIAL
- http://www.w3schools.com/graphics/svg_intro.asp

LENGUAJES: UIML, MXML, FXML

- UIML
- https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uiml
- MXML
- <https://en.wikipedia.org/wiki/MXML>
- FXML
- http://docs.oracle.com/javafx/2/fxml_get_started/jfxpub-fxml_get_started.htm