Tammam Kamer Aldeen

دورة في لغة XML

إعداد : تمام قمر الدين

للمزيد من الدورات والمعلومات زوروا صفحتنا

Database and website programming in your hands

www.facebook.com/yourcommands

**** الفهرس ****

الموضوع	رقم الصفحة
الإهداء	3
مقدمة تعريفية باللغة - ماذا تعنيXML	4
كيف تعمل لغة XML ؟؟؟؟	5
شجرة XML	6
قواعد الكتابة في لغة XML !!!!	9
عناصر XML	11
قواعد تسمية التاغات في XML	13
الواصفات Attributes في XML !!!!	15
المقارنة بين العناصر elements والواصفات Attributes !!!!	16
فضاءات الاسماء في XML !!!!	19
الحل لمشكلة تضارب الاسماء	20
ترميز XML !!!!	23
ماهو Unicode	23
الأخطاء في XML	24
عرض ملفات XML !!!!	25
عرض ملفات XML بصيغة CSS !!!!	26
أنواع الملفات في XML	29
تعريف أنواع المستندات DTD - Document Type Definition !!!!	32
استعراض البيانات - XMLHttpRequest بالجافا سكريبت	34
محلل XML – الجافا سكريبت	35
XML DOM - الوصل المعياري للمستندات	36
قراءة كامل ملف XML باستخدام حلقة For - بالجافا سكريبت	40
الكتابة في ملف XML باستخدام السي شارب	44
القراءة من ملفات XML – باستخدام السي شارب و VB.NET	47

الإهداء

أهدي هذا الكتاب الصغير إلى كل من ساعدني وشجعني على كتابته وكل من كان سببا في معلوماتي بعالم الحاسب من أهل وأصدقاء ومدرسين.

كما أهدي الكتاب اهداء خاصة لأمي وابي وملكة قلبي خطيبتي الغالية.

أصدقائي العلم جيد جدا وعلينا كل يوم أن نتعلم المزيد ولا نستسلم لأن كل واحد فينا يبرع في مجال معين وكل انسان يجب أن يبحث عما يرضي داخله من معارف.

الجهل أسوء طريقة ليكون فيها الإنسان غارقا في مطبات الحياة التعيسة لذا أصدقائي داوموا على قراءة الكتب والتعلم من مختلف المصادر، فليس من المعيب أن تسأل من هو أصغر منك سناً وأعلى منك شئنا علميا.

لا تقل غدا سأقرأ الكتاب الفلاني وبعده سأتدرب على التقنية أو بالمجال الفلاني، بل كن السابق وابدأ من اليوم فليس من الحكمة تأجيل عمل اليوم إلى الغد.

لن أطيل عليكم وسأبدأ كتابي مباشرة مع تمنياتي لكم بالتوفيق.

دورة في - XML الدرس الأول:

مقدمة تعريفية باللغة - ماذا تعني XML ؟

هي اختصار لـ eXtensible Markup Language وقد تم تصميمها لتناقل وتخزين البيانات ،وتعتبر من أكثر اللغات شعبية في عالم البرمجة وكما أنها سهلة التعلم.

وهنا مثال حول ملف من هذه اللغة:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to> Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

لاحظوا أن هذه اللغة تتميز ببنيتها الهرمية فهي عبارة عن عقدة رئيسية note ومجموعة من العقد الفرعية بداخلها مثل. to,from,heading,body .

إن لغة XMLهي لغة مصممة لتخزين البيانات وليست لعرضها ففي المثال السابق قمنا بتخزين بيانات ملاحظة note بتحديد المرسل إليه ثم المرسل ثم عنوان الملاحظة ونص الملاحظة كما هو مبين بالمثال.

لعرض هذه البيانات نحن بحاجة للغات رديفة لها مثل HTML,JavaScrite كما يمكن استخدام لغات أخرى مثل السي شارب والفي بي مع ASP.NET .

اما من حيث النص البرمجي فنجد أن هذه اللغة مشابه ل HTML ولكنهما مختلفتان كثيرا من حيث الأهداف :

- ان لغة XML تم تصميمها لتخزين البيانات وتحديد انواعها أما HTML فقد تم تصميمها لعرض البيانات والاهتمام بتنسيقها وشكل العرض.
- وفي المثال السابق لاحظنا أن XML لا تفعل شيء محدد سوى التخزين حيث أن عملها مشابه لعمل قواعد البيانات، فالمعلومات السابقة هي مجرد تخزين لملاحظة مرسلة من شخص لشخص مع تفاصيل الملاحظة ومع ذلك هذه اللغة لا تفعل شيء محدد سوى توصيف وتخزين البيانات أي أنها لا تقوم بعرض وغير مسؤولة عن عمليات التنسيق وغيرها.
- من المهم معرفة شيء عن هذه اللغة أنها مصممة لتقوم ببناء واختراع التاغ الذي ترغبه أنت <>ما بين هذه الأقواس والمقصود بذلك .

مثلا التاغ (Tag) المسمى <note>أو حتى <To>أو <From>هي ليست تاغات محجوزة مسبقا ضمن الكلمات المحجوزة باللغة حيث يمكنك وضع الاسماء التي ترغب بها وفيها بعد يمكنك الاستعلام من هذه التاغات كما ترغب.

- كما أن هذه اللغة ليست بديل HTML أبدا كما أسلفنا، وانما دعم لها لتخزين البيانات وهي موصى بها من w3C في February 10, 1998 .

دورة في XML - الدرس الثاني الجزء الأول

كيف تعمل لغة XML ؟؟؟؟

كما نعلم عند استخدامنا لغة HTML في بناء مواقع الانترنت وعندما نريد أن يكون الموقع ديناميكي متجدد فيتوجب علينا أن نقوم بتعديل كود HTML بشكل مستمر كلما تغيرت البيات أو قضت الحاجة لذلك. لكن بوجود XML ومع القليل من كود الجافا سكريبت نجد أن العملية أسهل بكثير ولسنا بحاجة لوقت وجهد تبديل المعلومات.

في عالم الحاسب كما نعلم وفي قواعد البيانات يتم تخزين البيانات على أنها بيانات تحمل أنماط معطيات مختلفة ومعقدة أحيانا مثل int, char, image وغيرها لكن في XML الأمر يختلف فكل البيانات يتم حفظها بالتنسيق النصي الاعتيادي مما يتيح أمام برامج الوصول إلى ملفات XML مشاركة لبيانات الملف بشكل سهل جدا.

ومن الملاحظ كثيرا أن أغلب لغات البرمجة تحتوي على مكتبات خاصة للتعامل مع XML مما يدل على أهمية هذه اللغة.
من أشهر استخدامات هذه اللغة لـ :

- XHTML .
- WSDL : لتوصيف خدمات الويب.
- web services.
- WAP وWML : كلغة ترميز للأجهزة المحمولة handheld devices.
- RSS : لتوصيف RSS.
- SMIL : لتوصيف ملفات الملتيميديا التي يتم تحميلها لموقع ويب ما.

دورة في XML - الدرس الثاني الجزء الثاني

شجرة XML !!!!

كما سبق وأسلفنا أن بينة ملفات هذه اللغة هي هرمية أو شجرية حيث أن هنالك شيء داخل شيء أو عقدة داخل عقدة داخل عقدة وهكذا.

فكما بالمثال الذي مر معنا في الدرس الأول :

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

سنبدأ بشرح الأسطر سطر سطر:

- السطر الأول :

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

هذا السطر يقوم بتعريفنا بأن الملف هو ملف xml ويقول لنا أن اصدا الـ xml هو 1.0 ، كما يبين لنا أن لغة الترميز هي UTF-8 .
لاحظ بدأ التعريف الخاص بملف XML بإشارة أصغر يليها اشارة استفهام وانتهى بإشارة استفهام واشارة أكبر .

- السطر الثاني :

هو العقدة الجذر <note> أي الـ Root الخاص بالملف حيث أنها عقدة اساسية بالملف ممكن أن يكون تسمية هذه العقدة باسم شامل لمحتوى الملف كأن تكون تمثل أن الملف الخاص بي هو ملاحظات من شخص لشخص لذا قمنا بتسميتها note .

- السطر الثالث والرابع والخامس والسادس:

هي عبارة عن العقد الأبناء للعقدة الجذر to, from, heading, and body .

- السطر السابع والأخير:

هو تأشيرة الإغلاق الخاصة بالعقدة الجذر مع ملاحظة أن كل تأشيرة يتم فتحها ما بين اشارتي أكبر وأصغر يتم إغلاقها بنفس الأشارتين ولكن مسبوقا الاسم بالإشارة / كما يلي مثلا <note/>

إذا : كم رأينا أن لغة XML هي لغة سهلة التوصيف وتتيح للمستخدم تحديد خياراته بنفسه.
قواعدها التي تحددها هي :

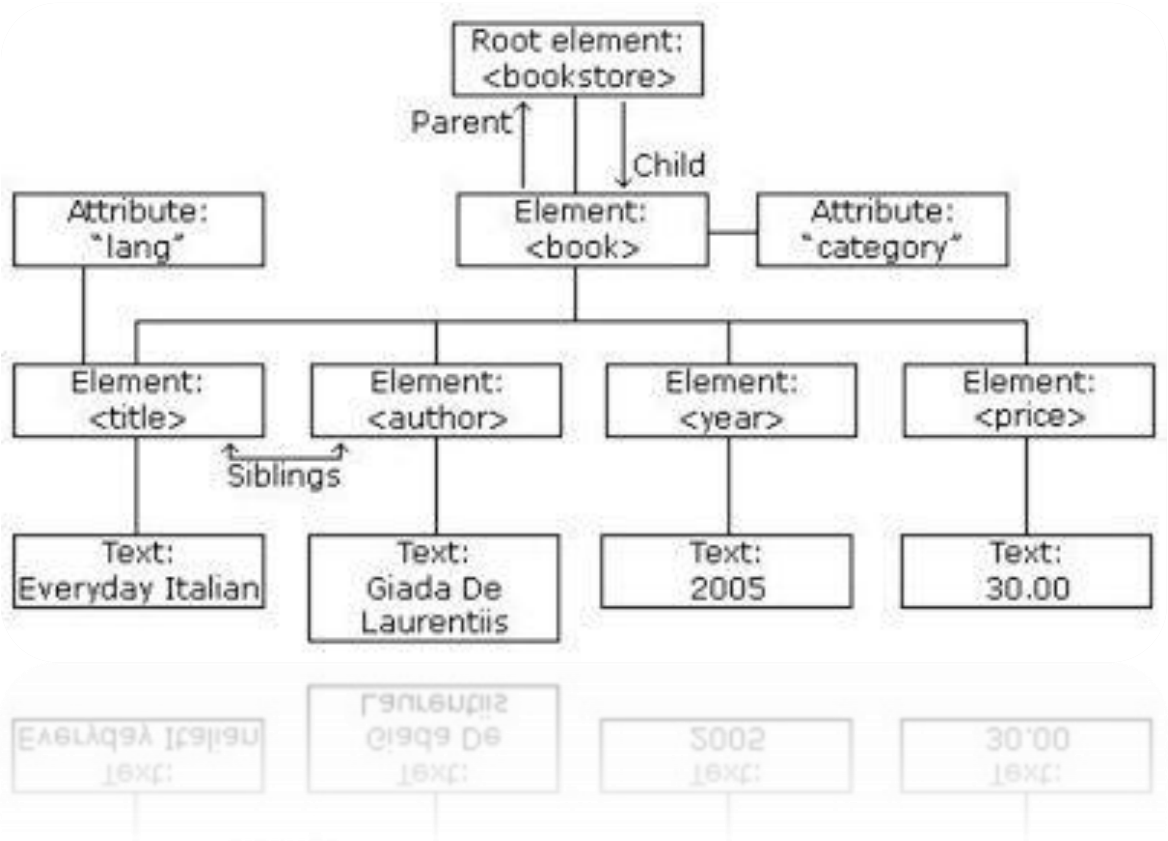
1- يجب أن يكون لدينا عقدة جذر أو عقدة أب.
2- كل العقد الأخرى هي عقد أبناء لهذه العقدة الجذر.

```
<root>
<child>
<subchild>.....</subchild>
</child>
</root>
```

وهنا مثال عملي عن بعض أسطر من ملف XML :

```
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

وأخيرا الصورة التالية تقوم بتوصيف هذا المثال العملي:


Root element:
<bookstore>
Parent
Child
Attribute:
"lang"
Element:
<book>
Attribute:
"category"
Element:
<title>
Element:
<author>
Element:
<year>
Element:
<price>
Siblings
Text:
Everyday Italian
Text:
Giada De
Laurentiis
Text:
2005
Text:
30.00

دورة في XML - الدرس الثالث الجزء الأول

قواعد الكتابة في لغة XML !!!!

XML صحيح أنها ليوجد فيها كلمات مفتاحية تحددها إلا أن هنالك بعض القواعد التي يجب على المبرمج اتباعها أثناء كتابته بهذه اللغة.

- القاعدة الأولى :

دائما أي تاغ Tag يتم فتحه في XML يجب أن يتم اغلاقه وإلا انت أمام خطأ فادح فعلى عكس HTML التي لا تجبرك على الإغلاق لتاغ قمت بفتحه مسبقا.
لاحظ في XML كيف يكتب التاغ

```
<note>
</note>
```

أما في HTML يمكن كتابته هكذا:

```
<p> some thing .
```

- القاعدة الثانية :

التاغ TAG في XML هو حساس لحالة الأحرف أي انك عندما تكتب بحرف أو أحرف كبيرة هي ليست مثلما وأنت تكتب بحرف أو أحرف صغيرة ويجب أن يكون تاغ الفتح والإغلاق بنفس حالة الأحرف، في المثال التالي نلاحظ خطأ ممكن أن يحصل مع أي شخص يكتب بهذه اللغة:

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

الأولى خاطئة والثانية صحيحة.

- القاعدة الثالثة :

يجب أن تكون تاغات الفتح والإغلاق منسقة ومرتبة وغير متداخلة وأن تكون العقد فيها منتظمة، ففي Html يمكن لنا أن نفتح تاغ معين ونضع بداخله تاغ آخر نغلق التاغ الأول ثم نغلق الثاني كما يلي:

```
<b><i>This text is bold and italic</b></i>
```

وهي حالة صحيحة في HTML أما في XML هذا غير صحيح وينتج عنه خطأ لذا يجب مراعاة التنسيق كما يلي:

```
<b><i>This text is bold and italic</b></i>
```

وهذا مثال XML على ذلك

```
<root>
<child>
```

```
<subchild>.....</subchild>
</child>
</root>
```

- القاعدة الرابعة :

قيم الخصائص في تاغات XML يجب أن يتم وضعها ضمن اشارات تنصيص " " لذا كن حريصا عليها، حيث أن الخصائص في XML هي مشابهة لـ HTML وهذا مثال يوضح الفكرة السابقة:

```
<note date="12/11/2007">
<to>Tove</to>
<from>Jani</from>
</note>
```

ضمن التاغ note لاحظ الخاصية date تم فيها وضع تاريخ ضمن اشارات تنصيص " " وهذا ضروري جدا وإلا هنالك خطأ.

- القاعدة الخامسة :

يجب احترام بعض المحارف الخاصة في XML وعدم استخدامها ضمن اللغة لأنها تولد أخطاء وهذه المحارف هي اشارة < و > و & و " و ' حيث تم الاستعاضة عن هذه المحارف في XML برموز أخرى يجب عليك استخدام هذه الرموز لو أردت استخدام هذه المحارف وهي كما يلي:

<	<
>	>
&	&
'	'
"	"

مع العلم أن استخدم < و & هم فقط الغير قانوني بهذه اللغة ولكن من الجميل استعاضة كل المحارف الخاصة الأخرى بالرموز الخاصة بها، ويأتي حجز هذه المحارف كونها مستخدمة ببناء تاغات اللغة.

القاعدة السادسة :

يتم كتابة لتعليقات في XML كما يلي :

```
<!-- This is a comment -->
```

حيث أن التعليق هو عبارة عن شرح لا يقوم المترجم الخاص باللغة بقراءته أو ترجمته كونه خاص بالمستخدم يشرح شيء معين عن الكود وهو ليس ضمن الكود.

-القاعدة السابعة :

المسافات بين الكلمات يتم اقتطاعها في XML تلقائيا لو كانت بمقدرا أكثر من مسافة واحدة بين الكلمتين على عكس HTML التي تحافظ على المسافات كما كتبها المستخدم فمثلا في HTML يمكننا كتابة التالي :

XML Learning

لاحظ وجود المسافات بين الكلمتين أما في XML لو حاولت كتابة النص السابق فسيظهر لك كما يلي:

XML Learning

بوجود مسافة واحدة بين الكلمتين.

دورة في XML - الدرس الثالث الجزء الثاني

عناصر XML !!!!

العناصر في XML هي كل شيء ما بين التاغات سواء كانت تاغات أخرى أو نصوص أو حتى واصفات للتاغات أو مزيج من العناصر التي سبق.
وليكن لدينا المثال التالي في XML:

```
<bookstore>
<book category="CHILDREN">
<title>Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title>Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

ان <bookstore> و <book> تحوي بداخلها مجموعة من العناصر وذلك لأنها تحتوي عناصر من تاغات ونصوص .
كما أن التاغ <book> يحتوي على الواصفة

```
category="CHILDREN"
```

وكذلك <title>، <author>، <year>، <price> تحتوي على نصوص وهي عناصر.
ملاحظة : يمكن للتاغ أن لا يحتوي بداخله شيء ويكون ذو قيمة فارغة مثل:

```
<book/><book>
```

ويمكن كتابتها بالشكل التالي <book /> أي اشارة / تأتي بعد اسم التاغ وليس قبله وتمسى هذه الطريقة بالإغلاق الذاتي للتاغ.

- قواعد تسمية التاغات في XML:

هذه الفقرة مهمة جدا في درسنا حيث أن التاغ يجب أن يتم كتابته بطريقة مفهومة وسلسلة وهنالك مجموعة قواعد متفق عليها يجب اتباعها أثناء كتابتن للكود:

- يمكن أن يحتوي اسم التاغ على نصوص أو أرقام أو أي محارف خاصة أخرى.
- لا يمكن للتاغ أن يبدأ اسمه برقم أو علامة ترقيم .
- لا يمكن أن يبدأ اسم التاغ بأي شكل من أشكال الأحرف التالية xml سواء بأحرف كبيرة أو صغيرة .
- لا يمكن أن نقوم بوضع فراغات ما بين الكلمات لو كان التاغ مكون من أكثر من كلمة لذا يفضل عندها وضع _ أو الكلمتان متلاصقتان مع بدأ الكلمة الثانية بحرف كبير.

لذا عزيزي المبرمج وزميلي في الصفحة قم باختيار اسماء التاغات بكشل يوصف التاغ ومحتواه مثل <first_name>، <last_name> لتعبر عن الاسم الأول والأخير.

وحاول أن يكون الاسم قصير ومختصر قدر الإمكان فهو اسهل للقراءة وترتيب بياناتك فاكتب مثلا <book_title> بدلا من :

<the_title_of_the_book>

- لا تحاول استخدام اشارة الناقص - لأن بعض البرمجيات قدد تفهم من كودك أنك تريد طرح شيئين من بعضهما وهو غير صحيح عندما نريد توصيف الاسم الأول مثلا كما يلي first-name

- تجنب أيضا استخدام النقطة . لأن بعض البرمجيات تظن أن الكلمة الثانية هي خاصية من خواص الكلمة الأولى first.name أي الكلمة name هي احد خواص الغرض first وهو غير صحيح.

- تجنب استخدام النقطتين : لأنها مستخدمة في استيراد فضاءات الاسماء في XML .

- استخدامك لأحرف غير الأنكليزية مثل éòá هو قانوني ولكن قد يسبب بعض المشاكل بالخرج لذا ابتعد عنها.

لغة XML هي لغة قابلة للتوسع !!!!

بفرض أنك تريد ارسال ملاحظة note وليكن كود XML الخاص بنا كما يلي:

```
<note>
<to>Tove</to>
<from>Jani</from>
<body>Don't forget me this weekend!</body>
</note>
```

وفيما بعد أراد المبرمج أن يقوم بتوسيع البيانات التي سيتم ارسالها بالتطبيق لتصبح كما يلي:

```
<note>
<date>2008-01-10</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

اي أنه اضاف التاغ Date لملف XML الخاص به فهل هنا تظهر رسالة خطأ ويتم انهاء التطبيق دون نتيجة؟؟؟؟

الإجابة هي: لا فكما قلنا XML قابلة للتوسع ويمكن لنا أن نضيف عليها البيانات التي نرغب مستقبلا وتوسعتها كما نريد.

الواصفات Attributes في XML !!!!

إن التاغات في XML لها واصفات Attributes مثلها مثل التاغات في HTML وتعمل هذه الواصفات على تزويدنا بمعلومات اضافية عن العناصر elements .
ليكن لدينا المثال التالي في HTML :

```
<img src="computer.gif">
```

تعتبر src هي واصفة من واصفات img حيث تزودنا بمسار الصورة التي نرغب بعرضها.
وكذلك في XML هي تعمل على تزويدنا بمعلومات اضافية عن العناصر الخاصة بالواصفة كما يلي:

```
<file type="gif">computer.gif</file>
```

فالواصفة type تحمل القيمة gif لتخبرنا أنا نوع الملف الذي قمنا بحفظ اسمه ضمن تاغ xml هو من نوع gif .

ملاحظة: يجب وضع قيمة الواصفة ضمن اشارات تنصيص مفردة او مزدوجة هكذا " " أو هكذا ' '.

```
<person sex="female">
<person sex='female'>
```

فكلا الحالتين صحيحتين.

فإذا كانت قيمة الواصفة مثلا تحتوي اشارات التنصيص المزدوجة يمكنك استخدام اشارات التنصيص المفردة لاحتواء كامل قيمة الواصفة كما في المثال التالي:

```
<gangster name='George "Shotgun" Ziegler'>
```

كما يمكن الاستعاضة عن اشارات التنصيص السابقة بالبديل الخاص بها كما يلي:

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

المقارنة بين العناصر elements والواصفات Attributes !!!!

ليكن لدينا المثالين التاليين:

- الأول :

```
<person sex="female">
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

- الثاني :

```
<person>
<sex>female</sex>
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

قارن بنفسك بينهما بالشكل بداية ماذا تجد ؟؟؟؟
في المثال الأول قمنا بتعرف الجنس sex على أنه واصفةAttribute أما في المثال الثاني استخدمناه كعنصر element ولا يوجد قاعدة تحدد لنا متى يمكن استخدام الواصفات ومتى يمكن استخدام العناصر، فقم انت باختيار الطريقة التي تفضلها وقم باستخدامها.

سأقوم الآن بطرح مثال ولكن بثلاث طرق لتفهموا بشكل أوضح ماهي xml وماهي الخيارات المتاحة فيها.
الأول: وفيه نستخدم التاريخ كواصفة

```
<note date="10/01/2008">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

الثاني: وفيه نستخدم التاريخ كعنصر

```
<note>
<date>10/01/2008</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
```

```
<body>Don't forget me this weekend!</body>
</note>
```

الثالث: وفيه نفصل التاريخ كعناصر اي اليوم عنصر والشهر عنصر أخر والسنة عنصر آخر

```
<note>
<date>
<day>10</day>
<month>01</month>
<year>2008</year>
</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

وأنا شخصيا أفضل استخدام الطريقة الثالثة اي استخدام العناصر elements وتفصيل العناصر عن بعضها البعض كما في المثال الثالث السابق والسبب يعود لما يلي :

- الواصفات Attribute صعبة التوسع في المستقبل.
- الواصفات Attribute لا يمكن أن تحتوي قيم مركبة من أكثر من كلمة.
- الواصفات Attribute لا تملك البنية الشجرية التي تملكها العناصر.

أحد أبرز استخدامات الواصفة Attribute هي استخدامها لحفظ ID الكتلة التي نقوم بتخزينها كما في المثال التالي :

```
<messages>
<note id="501">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
<note id="502">
<to>Jani</to>
<from>Tove</from>
<heading>Re: Reminder</heading>
<body>I will not</body>
</note>
</messages>
```

دورة في XML - الدرس الخامس

فضاءات الاسماء في XML !!!!

عادة ما نقوم في XML بوضع الكثير من التاغات Tags في أكوادنا ونقوم باختيار اسماء معينة لهذه الأكواد .

ولكن وفي بعض الأحيان قد نكون اخترنا لاسم تاغ في الكود مشابه لاسم تاغ آخر في HTML مما يسبب لنا الإرباك في الكود وقد ينتج عنه خطأ، لنستوضح ذلك في مثال.

ليكن لدينا كود HTML التالي :

```
<table>
<tr>
<td>Apples</td>
<td>Bananas</td>
</tr>
</table>
```

هو عبارة عن جدول يحوي خليتين وفيها بياناتهم الخاصة بهم.

وليكن لدينا كود XML التالي:

```
<table>
<name>African Coffee Table</name>
<width>80</width>
<length>120</length>
</table>
```

وهو كود اعتيادي قمنا بكتابته بقواعد كتابة XML النظامية.

الأكواد فوق متشابه كون الكود الأول هو جدول table يبدأ التاغ الخاص به بهذا الكود والثاني كود HTML قمنا بكتابته واختيار الاسم table للعقدة الجذر فيه.

نحن الآن قمنا بالشرح وفهمنا ذلك ولكن التطبيقات المسؤولة عن القراءة كيف لها أن تفهم ذلك؟؟؟؟

الحل لمشكلة تضارب الأسماء !!!!!

الحل بسيط جدا حيث يمكننا أن نقوم بتمييز الأسماء لعناصر XML من خلال وضع بادئة لهذا الكود، أنت تتساءل ماذا يعني بادئة ببساطة هي بالانكليزية Prefix ولشرح كيف نقوم بها يمكنك الإطلاع على المثال التالي :

```
<h:table>
<h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
</h:table>

<f:table>
<f:name>African Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>
```

بهذه الطريقة يمكن تمييز كود HTML عن كود XML بكل بساطة.

لاستخدام الطريقة السابقة عليك تعريف فضاء الأسماء الخاص بذلك حيث يتم ذلك ضمن التاغ الأول او الجذر الذي بدأ فيه استخدام هذه البادئة حيث يمكن استدعاء فضاء الأسماء السابق من خلال xmlns كما يلي:

```
xmlns:prefix="URI"
```

الآن إليكم المثال التالي:

```
<root>

<h:table xmlns:h="http://www.w3.org/TR/html4/">
<h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
```

```
</h:table>

<f:table xmlns:f="http://www.w3schools.com/furniture">
<f:name>African Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>

</root>
```

في هذا المثال، الواصفة xmlns في التاغ <table> أخذت القيمتين h: و f: النقطتان بعد الحرف، هنا تم اعطاء هذين الحرفين لتأهيل البادئتين ليتم استخدامهما في كود XML .

مع العلم أنه بما أننا قمنا بتعرف بادئة لعنصر أب فإن كافة العناصر الأبناء تأخذ هذه البادئة عند التعريف.
كما يمكن أن يتم التصريح عم البادئة (xmlns) في المكان الذي نرغب باستخدامه فيها أو ضمن العقدة الجذر.

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">

<h:table>
<h:tr>
<h:td>Apples</h:td>
<h:td>Bananas</h:td>
</h:tr>
</h:table>

<f:table>
<f:name>African Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>

</root>
```

ملاحظة : استخدام الرابط URI ضمن تعريف البادئة هو ليس لأغراض برمجية وللقيام بعمليات تحليل ضمن مترجم اللغة ،وانما هو فقط لاعطاء اسم فريد لفضاء الأسماء الذي نقوم بتعريفه.

- ماهو URI سأقوم بالشرح هو اختصار لـ Uniform Resource Identifier أي معرف الموارد الموحد وهو عبارة عن سلسلة من المحارف التي تحدد مصادر الانترنت.

دورة في XML - الدرس السادس الجزء الأول

ترميز XML !!!!

يمكن لملفات XML أن تحتوي على أحرف للغات عالمية مختلفة مثل العربية والإنكليزية والنرويجية æøå والفرنسية êèé وغيرها من اللغات المعترف عليها في مواثيق البرمجة والحوسبة.

لذا لتجنب أي أخطاء صادرة عن سوء الترميز يجب علينا تحديد ترميز للمحارف التي يتم ادخالها أو حفظ ملف XML الخاص بنا على أنه UTF-8 ....

تزميز لمحارف يعمل على تحديد رمز ثنائي binary code فريد لكل المحارف الغريبة التي يتم استخدامها وتخزينها في هذا الملف.

ويمكن أن نطلق أيضا على ترميز المحارف تموضع المحارف character set أو حتى خريطة المحارف character map أو تموضع الكود code set أو صفحة الكود code page .

ماهو Unicode ؟؟؟؟

هو معيار خلق ترميز المحارف لمستند نصي ما، حيث يقوم بتعريف أغلب اللغات العالمية من خلال اعطائها اسم ورقم خاص بهذه اللغة.

وتحتوي Unicode على متغيرين اثنين هما UTF-8 و UTF-16 .

- حيث أن UTF هي اختصار لـ Universal character set Transformation Format

- UTF-8 : تستخدم بايت واحد (ثمانية بيتات) للمحارف الأكثر شيوعا، و بايتين أو ثلاثة لبقية المحارف وهي الأساسية Standard لشبكة الويب وهي الترميز الافتراضي لـ HTML-5, CSS, JavaScript, PHP, SQL, XML.

- UTF-16 : تستخدم بايتين للمحارف الأكثر شيوعا وثلاثة لباقي المحارف.

أين يمكن أن أقوم بالتعريف على الترميز الذي أرغب ؟؟؟؟

في السطر الأول من مستند XML

```
<?xml version="1.0"?>
```

يمكن أن نقوم بإضافة ما يلي "encoding="UTF-8 ليصبح السطر كما يلي :

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

إن كافة التطبيقات التي تستخدم لقراءة تطبيقات XML افتراضيا تستطيع قراءة الترميز UTF-8 و UTF-16 .

وفي حال لم نقم بذكر الترميز فإن UTF-8 تكون هي الافتراضية ، ومن الجدير بالذكر ان الكثير من التطبيقات التي تستخدم XML يمكنها أن تفهم الترميزات التالية :

ISO-8859-1,
Windows-1252,
ASCII

الأخطاء في XML !!!!

في معظم الأحيان ان مستندات XML يتم انشاءها يكون على كمبيوتر واحد أما تعديلها ورفعها إلى السيرفر يكون على كمبيوتر آخر ويتم عرضها من خلال المتصفح في كمبيوتر آخر.

فإذا كان الترميز على الكمبيوترات الثلاثة غير صحيح فإن المتصفح يقوم بعرض المحارف بشكل غير مفهوم أو يقوم بعرض رسالة خطأ.

دورة في XML - الدرس السادس الجزء الثاني

عرض ملفات XML !!!!

يمكن أن يتم عرض ملفات XML في كل المتصفحات في العالم تقريبا، ولا تتوقع أبدا أن يكون تنسيق هذه الملفات أبدا مثل تنسيق ملفات HTML .

لأنها تبدو كما النص البرمجي تقريبا، فالكود التالي يبدو في المتصفح كما في الصورة المرفقة :

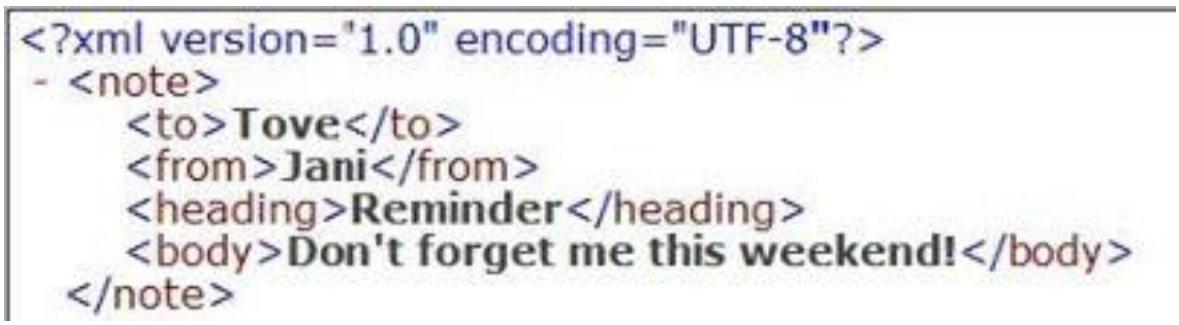
```
<?xml version="1.0" encoding="UTF-8"?>
- <note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

حيث أنه يتم تلوين اسم العقد في المتصفح ويتم اضافة اشارة + و - ليتم تمديد وفتح الكود او تصغيره أي expand و collapse .

ولعرض النص الاساسي لملف XML بدون تلوينات أو اشارات + و - قم بعرض مصدر الصفحة التي أنت تقوم بفتحها.

ويتم العرض بهذه الطريقة كون XML لا تهتم بطريقة عرض البيانات وانما كما قلنا هي مخزن بسيط للبيانات .

وفيما يلي رابط لثلاث ملفات XML لتتمعنوا فيها جيدا.
http://www.4shared.com/rar/0Xcb09Mzce/XML_Example.html



دورة في XML - الدرس السادس الجزء الثالث

عرض ملفات XML بصيغة CSS !!!!

من الممكن أن نقوم باستخدام CSS لعرض ملفات XML الخاصة بنا.
ليكن لدينا ملف XML التالي :

```
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
<CD>
<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>
</CD>
</CATALOG>
```

وليكن لدينا ملف CSS التالي :

```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```

لربط ملف CSS مع ملف XML نقوم بإضافة السطر التالي في ملف XML :

```
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
```

ليكون شكل أول أربع أسطر من ملف XML كما يلي:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
<CD>
```

ويكون شكل العرض كما يلي:

Empire Burlesque Bob Dylan
USA
Columbia
10.90
1985

Hide your heart Bonnie Tyler
UK
CBS Records
9.90
1988

Greatest Hits Dolly Parton
USA
RCA
9.90
1982

دورة في - XML الدرس السابع الجزء الأول

أنواع الملفات في XML !!!!

عندما نقوم بكتابة ملف XML معين ويكون كل التعليمات فيه صحيحة يسمى هذا الملف Well formed.

وللتذكير فقط الملف الصحيح البنية في XML يجب أن يحقق القواعد التالية:

- يجب أن يحتوي ملف XML العنصر الجذر root .
- كل عنصر من عناصر XML يجب أن يكون له تاغ للإغلاق.
- تاغات XML هي حساسة لحالة الأحرف.
- العناصر يجب أن تكون محتوات بعضها ضمن بعض nested .
- قيم الواصفات في XML يجب أن يتم وضعها ضمن علمتي التنصيص" " .

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

اليوم درسنا ليس كالمعتاد قمنا بذكر قواعد كتابة XML وقمنا بتزويدكم بكود XML وأقدم لكم رابط لموقع يعمل كمصحح لأكوادXML .

وهذا هو الرابط:

http://validator.w3.org/#validate_by_input

الآن عليكم الدخول للرابط السابق وكتابة كود XML الخاص بكم والتحقق من النتيجة.
كما أنني سأقوم بتزويدكم بوظيفة قوموا فيها باختبار مهاراتكم الشخصية، الوظيفة هي تصحيح الأكواد التالية:

الكود الأول:

```
<?xml version="1.0" encoding="UTF-8"?>
<note/>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

الكود الثاني:

```
<?xml version="1.0" encoding="UTF-18"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

الكود الثالث:

```
<?xml version=1.0 encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

الكود الرابع:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</Body>
</note>
```

قم بزيارة الرابط المرفق وقم بتدقيق الأكود السابقة ولكن قبل ذلك أتمنى أن تحاولو تدقيق الأكواد بأنفسكم ومن دون غش، ثم ان فشلتم ستجدون الحل بالرابط المرفق على الرابط السابق.

دورة في XML - الدرس السابع الجزء الثاني

تعريف أنواع المستندات DTD - Document Type Definition !!!!

عندما يملك ملف XML تاغات مكتوبة بشكل صحيح نقول أنه Well Formed وكذلك المستندات تكون متوافقة مع قواعد كتابة أنواع المستندات DTD .

مثال :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ماذا تعني هذه الصيغة؟ ماذا يعني هذا الكلام ؟ تابعوا معي !!!!

- DOCTYPE note! : يحدد أن هذا العنصر note هو العنصر الجذر في الوثيقة.
- ELEMENT note! : يقوم بتعريف أن الجذر note يحتوي على العناصر التالية : to,from,heading,body
- ELEMENT to : يحدد أن هذا العنصر هو من النوع PCDATA .
- ELEMENT from : يحدد أن هذا العنصر هو من النوع PCDATA .
- ELEMENT heading : يحدد أن هذا العنصر هو من النوع PCDATA .
- ELEMENT body : يحدد أن هذا العنصر هو من النوع PCDATA .

ملاحظة PCDATA: تعني أنه نص قابل لتحليل محتواه لأنواع بيانات.
parse-able text data

كما أن الترويسة DocType تصلح لأن تستخدم لكتابة محتوى ملف DTD كما يلي:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
<!ENTITY nbsp " ">
<!ENTITY writer "Writer: Donald Duck.">
```

```
<!ENTITY copyright "Copyright: W3Schools.">
]>

<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<footer>&writer; &copyright;</footer>
/>note<
```

لاحظ معي في العنصر Footer وجود اشارة &ثم اسم الكيان مثلا writer ثم فاصلة منقوطة ؛ ولاحظ في محتوى DocType في نص DTD عرفنا الكيانات التالية nbsp و writer و copyright حيث أن استدعاء الكيان يتم بالطريقة السابقة & ثم اسم الكيان ثم الفاصلة المنقوطة ؛ عند تنفيذ الملف السابق وعرضه تظهر النتيجة كما يلي :

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<footer>Writer: Donald Duck. Copyright: W3Schools.</footer>
</note>
```

دورة في XML - الدرس الثامن الجزء الأول

استعراض البيانات - XMLHttpRequest بالجافا سكريبت

يتم استخدام الغرض XMLHttpRequest لتبادل البيانات بين الزبون والسيرفر وهذا الغرض object له فواد عديدة :

- يقوم بالعديل على الصفحة بدون اعادة تحميل هذه الصفحة .
- يقوم بإرسال البيانات إلى السيرفر بعد أن يتم تحميل الصفحة لدى الزبون .
- يستقبل البيانات من السيرفر بعد أن يتم تحميل الصفحة لدى الزبون.
- ارسال البيانات إلى السيرفر يكون بالخلفية ودون أن يشعر المستخدم بذلك.

كيفية انشاء غرض من نوع XMLHttpRequest

كل المتصفحات الحديثة مثل IE7+, Firefox, Chrome, Safari, Opera تدعم عمل وانشاء أغراض من هذا النوع حيث تكون الصياغة العامة لإنشاء متحولات من هذا النوع كما يلي:

```
xmlhttp=new XMLHttpRequest();
```

أما بالنسخ القديمة من المتصفحات مثل IE6, IE5 فيتم انشاء بديل عن هذا الغرض وتكون الصياغة العامة له كما يلي :

```
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
```

حيث أن هذه المتصفحات لا تدعم أغراض XMLHttpRequest وانما ActiveXObject .

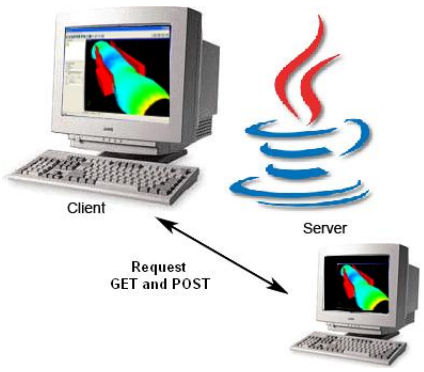
دورة في XML - الدرس الثامن الجزء الثاني

محلل XML - الجافا سيكريبت !!!!

إن معظم المتصفحات الحديثة تم فيها دمج محلل XML افتراضيا وتعمل هذا العمل بشكل تلقائي.
حيث يتم تحويل مستند XML إلى غرض XML DOM الذي يمكن فيه التعامل مع الجافا سيكريبت.

في المثال التالي سنقوم بشرح كيف يمكن لنا الاتصال مع ملف XML :

```
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
```



في البداية قمنا بالتأكد من نوع المتصفح فإذا كان حديثا قم بإنشاء XMLHttpRequest وقم بتخزينه ضمن المتحول xmlhttp وإلا قم بإنشاء ActiveXObject وقم بتخزينه ضمن المتحول نفسه xmlhttp .

فيما بعد قم بفتح الاتصال مع ملف XML :

```
xmlhttp.open("GET","books.xml",false);
```

وسيتم شرحه في درس لاحق.

ثم قم بإرسال طلب XMLHttpRequest ثم خزن الاستجابة في المتحول xmlDoc
بكل بساطة هكذا يتم الاتصال مع ملفات XML

دورة في - XML الدرس التاسع (تذكرة)

XML DOM - الوصل المعياري للمستندات

تحدد هذه الطريقة الوصول المعياري لمستندات XML والتعامل معها حيث أن تقوم بعرض ملفات XML ببنيتها الشجرية، مما يتيح الوصول لأي عنصر من العناصر بالشكل الشجري.

يمكن أن يتم اضافة عنصر أو خاصة أو التعديل والحذف عليهم بشكل سهل وسلسل جدا.

XML DOMهو باب واسع من أبواب XMl ولكن نحن هنا بصدد التعرف عليها بشكل عام ونأخذ منها احتياجاتنا بالمرحلة الحالية فقط.

الآن لنطبق مثال عملي ولتفهم أكثر قم بتحميل المرفقات ملف HTML مع ملف XML من الرابط التالي:

http://www.4shared.com/rar/3KC8DTHtce/Example__1_.html?

ملف HTML :

```
<html>
<body>
<h1>W3Schools Internal Note</h1>
<div>
<b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
</div>

<script>
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```

```
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>

</body>
</html>
```

ملف XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

أظن أنكم قادرين على قراءة الملف السابق ففي الجزء التالي من ملف HTML :

```
<b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
```

قمنا بتعريف span واعطينا كل واحد ID خاص به حتى نقوم لاحقا بالكتابة فيهم من خلال الجافا سكريبت.

في الجزء :

```
<script>

</script>
```

نقوم بكتابة الكود الخاص بنا بالجافا سكريبت وسأعيد شرح الكود خطوة بخطوة للتذكير:

```
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```

هنا نقوم بتعريف متحول xmlhttp نخزن فيه الطلب HTTP الذي سنقوم باستخدامه لإرسال واستقبال البيانات من خلاله هو المفتاح الذي يمكننا من الولوج للسيرفر وتنفيذ التعليمات الخاصة بنا.

```
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
```

هنا نقوم بإخبار المتحول xmlhttp أن يقوم بفتح الاتصال مع الملف note.xml في السيرفر ويستعد للقراءة ومن ثم نرسل الطلب للسيرفر وأخير نقوم بقراءة البيانات من ملف XML في response ونخزنها ضمن المتحولxmlDoc .

الآن المتحول xmlDoc يحوي كل البيانات الخاصة بملف XML يمكننا طلبها من خلاله. الآن وبالتعليمة التالية:

```
document.getElementById("to").innerHTML=

xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
```

نقوم بجلب البيانات من العقدة to في ملف XML ونقوم بكتابة محتوى العقدة ضمن عنصر HTML الذي قمنا بتعريفه وهو span والذي يحمل. ID=to

وكذلك في باقي العناصر from,message ،اذا لاحظنا كيف تتم عملية القراءة.

ملاحظة: التعليمة getElementsByTagName تقوم بالبحث عن التاغ الذي نقوم بإعطائها اسمه.

دورة في XML - الدرس العاشر

قراءة كامل ملف XML باستخدام حلقة For - بالجافا سكريبت

في المرفقات على الرابط التالي مثال كامل الشرح وكي لا تصبح الصفحة طويلة من كثرة الكود البرمجي سأقوم باختصاره والشرح عليه ويمكنكم أصدقائي تحميل المثال كاملا من الرابط :
http://www.4shared.com/rar/OSkKs0Kaba/CD_Catalog_Example.html

الآن سأبدأ بالشرح على الكود هنا جزء من ملف XML كما يلي:

```
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
</CATALOG>
```

أكيد أنتم لستم بحاجة لشرح هذا الكود فالمتابعين أصبحوا خبراء بقراءته وأقدم لكم كود HTML الخاص بهذا الكود :

```
<html>
<body>
<script>
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
```

```
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.write("<table border='1'>");
var x=xmlDoc.getElementsByTagName("CD");
for (i=0;i<x.length;i++)
{
document.write("<tr><td>");
document.write(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].
nodeValue);
document.write("</td><td>");
document.write(x[i].getElementsByTagName("TITLE")[0].childNodes[0].n
odeValue);
document.write("</td></tr>");
}
document.write("</table>");
</script>
</body>
/>html>
```

هذا ربما يحتاج للقليل من الشرح والشرح سيكون عند حلقة For تحديدا، فبعد تعريف المتحول xmlhttp وتخزين XMLHttpRequest بداخله وارسال بيانات الطلب إلى السيرفر من خلال:

xmlhttp.open
xmlhttp.send

وتخزين الإجابة responseXML في المتحول xmlDoc ،الآن سنبدأ الكتابة على ملف HTML الخاص بنا من خلال تعليمة :

```
document.write("");
```

إن مابين اشارات التنصيص " " كله سيتم كتابه على الصفحة سواء من نصوص أو تاغات HTML
نقوم بتعريف المتحول x ونخزن به العنصر الذي يحمل اسم التاغ CD :

```
xmlDoc.getElementsByTagName("CD");
```

حيث أن x هو الآن بمثابة مصفوفة تحوي بداخلها بيانات CD مرتبة بحسب التاغات حيث أن التاغ الأول artist رقمه صفر والتاغ الثاني TITLE رقمه واحد وذلك سيكون ضمن حلقة for فقط التي ستأتي لاحقا.

الآن نقوم بكتابة ترويسة بداية رسم الجدول على الصفحة كما يلي :

```
document.write("<table border='1'>");
```

ثم نبدأ بحلقة FOR

```
for (i=0;i<x.length;i++)
```

متحول البدء هو i ويبدأ بالقيمة صفر نقوم برسم tr و ثم Td على الصفحة

```
document.write("<tr><td>");
```

بالخطوة التالية سنكتب البيانات ومن ثم نغلق tr و td :

```
document.write(x[i].getElementsByTagName("ARTIST")[0].childNode
s[0].nodeValue);
```

الآن قمنا بكتابة محتوى لعقدة artist بعد أن قمنا بالبحث عن التاغ الذي يحمل ذلك الاسم. أما الآن علينا إغلاق tr الخاصة بنا وفتح td جديدة :

```
document.write("</td><td>");
```

من ثم علينا الكتابة ضمن td الجديدة وسنقوم بكتابة TITLE فيها كما يلي:

```
document.write(x[i].getElementsByTagName
("TITLE")[0].childNodes[0].nodeValue);
```

والآن نقوم بإغلاق td و tr معا كون لدينا جدول يحتوي على عمودين فقط وعدة أسطر وتستمر حلقة for بالقراءة من المصفوفة x عنصر يلي العنصر الآخر وهكذا حتى يتم قراءة كامل الملف وكتابة محتواه بالشكل السابق ضمن الصفحة لدينا.

وأخيرا بعد انتهائن من حلقة for لن ننسى إغلاق الجدول Table الخاص بنا بالتعليمة :

```
document.write("</table>");
```

وذلك حتى يكتمل التصميم، ولمن لم يفهم جيدا ماذا كان يجري أقول له عندما نريد بناء جدول يكون بالشكل التالي :

```
<table>
<tr>
<td></td>
<td></td>
</tr>
</table> </table>
```

نحن كل ما قمنا به هو استخدام التعليمة document.write لرسم هذا الجدول على الصفحة وقد قمنا برسمه سطرا سطرا، حيث قمنا بكتابة ضمن " " الخاصين بهذه التعليمة
السطر الأول <table> ومن ثم السطر الثاني كان ضمن حلقة for كما يلي:

```
<tr>
<td>
```

ثم أخذنا النص الذي سيكتب بالخلية من المصفوفة x ثم أغلقنا الخلية الأولى وفتحنا الخلية الثانية

```
<td><td/>
```

ثم نكتب النص المفترض من المتحول x وهو المصفوفة ومن ثم نغلق الخلية الثانية <td><tr/> والسطر كاملا لتعود حلقة for من جديد للدورا وترسم سطرا جديدا.

وبالنهاية نغلق الجدول <table/> .

ملاحظة أخيرة :

```
x[i].getElementsByTagName("ARTIST")[
```

هذه التعليمة تعني أعطني العنصر i من المصفوفة x والذي اسمه ("ARTIST")

دورة في XML - الدرس الحادي عشر

الكتابة في ملف XML باستخدام السي شارب

ليكن لدينا ملف XML يحمل الشكل التالي بالنسبة للعقد :

```
<records>
<record>
<name></name>
<Email></Email>
<Comment></Comment>
</record>
</records>
```

بحيث أن العقدة الجذر هي records فيها عقدة أساسية هي record وضمنها العقد الأبناء name, Email, Comment .

سنقوم بإعداد ثلاث مربعات نصية textbox مع زر حفظ كما في الصورة المبينة أدناه سنقوم بادخال الاسم والبريد الإلكتروني والملاحظة ثم ننقر على زر الحفظ save ليتم تخزين البيانات في ملف XML يحمل الاسم StoreUserInfo.xml

الموجود على أرضية السيرفر مباشرة (أي أنه غير مضمن داخل أي مجلد) عند النقر على الزر سيتم تنفيذ الكود التالي:

```
XmlDocument oXmlDocument = new XmlDocument();
oXmlDocument.Load(Server.MapPath("StoreUserInfo.xml"));
XmlNode oXmlRootNode = oXmlDocument.SelectSingleNode("records");
XmlNode oXmlRecordNode =
oXmlRootNode.AppendChild(oXmlDocument.CreateNode(XmlNodeType.
Element,"record",""));
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element,"name","")).InnerText=TextBox1.Text;
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element, "Email", "")).InnerText = TextBox2.Text;
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element, "Comment", "")).InnerText = TextBox3.Text;
oXmlDocument.Save(Server.MapPath("StoreUserInfo.xml"));
```

سأقوم بشرح هذا الكود بالتفصيل سطرا سطرا :

- السطر الأول :

```
XmlDocument oXmlDocument = new XmlDocument();
```

عرفنا متحول من نوع XmlDocument مهمته فتح ملف xml لنقوم بحفظ البيانات فيه لاحقا وهو المسؤول عن تطبيق التغيرات على ملف xml الفيزيائي.

- السطر الثاني :

```
oXmlDocument.Load(Server.MapPath("StoreUserInfo.xml"));
```

الذي قمنا بتعريفه oXmlDocument بالذاكرة ضمن المتحول xmlيقوم هذا السطر بتحميل ملف حتى بتسنى لنا التعامل مع هذا الملف من خلال هذا المتحول.

- السطر الثالث :

```
XmlNode oXmlRootNode =
oXmlDocument.SelectSingleNode("records");
```

قمنا بتعريف ماهي العقدة الجذر ضمن هذا الملف الحالي

-السطر الرابع :

```
XmlNode oXmlRecordNode =
oXmlRootNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element,"record",""));
```

قمنا بتعريف العقدة الابن الاساسية وهي التي تلي العقدة الجذر ،أي أنها العقدة التي تحتوي على مجموعة العقد name ,email, comment وهي بيانات المستخدم.
هنا نقوم بإنشاء هذه العقدة فارغة ونقوم بتهيئتها لنضيف إليها باقي العقد لاحقا.

- السطر الخامس :

```
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode
(XmlNodeType.Element,"name","")).InnerText=TextBox1.Text;
```

نقوم بإنشاء العقدة name ضمن العقدة record ونقوم بتزيدها بالبيانات الخاصة بها ونحصل على هذه البيانات من مربع النص الأول .

- السطر السادس :

```
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode
(XmlNodeType.Element, "Email", "")).InnerText = TextBox2.Text;
```

نقوم بإنشاء العقدة email ضمن العقدة record ونقوم بتزيدها بالبيانات الخاصة بها ونحصل على هذه البيانات من مربع النص الثاني.

- السطر السابع :

```
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode
(XmlNodeType.Element, "Comment", "")).InnerText = TextBox3.Text;
```

نقوم بإنشاء العقدة commentضمن العقدة record ونقوم بتزيدها بالبيانات الخاصة بها ونحصل على هذه البيانات من مربع النص الثالث.

- السطر الثامن :

```
oXmlDocument.Save(Server.MapPath("StoreUserInfo.xml"));
```

نقوم فيه بحفظ البيانات الحاصلة جميعا في ملف xml الموجود على السيرفر.

الآن نفس الخوات السابقة نقوم بتطبيقها بكود VB.NET ليكون الكود كما يلي :

```
Dim oXmlDocument As XmlDocument = New XmlDocument
oXmlDocument.Load(Server.MapPath("StoreUserInfo.xml"))
Dim oXmlRootNode As XmlNode =
oXmlDocument.SelectSingleNode("records")
Dim oXmlRecordNode As XmlNode =
oXmlRootNode.AppendChild(oXmlDocument.CreateNode(XmlNodeType.
Element, "record", ""))
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element, "name", "")).InnerText = TextBox1.Text
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element, "Email", "")).InnerText = TextBox2.Text
oXmlRecordNode.AppendChild(oXmlDocument.CreateNode(XmlNodeTyp
e.Element, "Comment", "")).InnerText = TextBox3.Text
oXmlDocument.Save(Server.MapPath("StoreUserInfo.xml"))
```

دورة في XML - الدرس الثاني عشر

القراءة من ملفات XML – باستخدام السي شارب و VB.NET

ليكن لدينا ملف XML يحمل الشكل التالي بالنسبة للعقد:

```
<records>
<record>
<name></name>
<Email></Email>
<Comment></Comment>
</record>
</records>
```

بحيث أن العقدة الجذر هي records فيها عقدة أساسية هي record وضمنها العقد الأبناء name, Email, Comment .

سنقوم بإعداد ثلاثة Label ليتم عرض البيانات فيهم.

تعلمنا بالدرس السابق كيفية الإضافة الآن سنتعلم كيفية القراءة من خلال عملية بحث عن اسم المستخدم name من خلال مربع نص لنبدأ بالكود كاملا :

```
DataSet ds = new DataSet();
      ds.ReadXml(Server.MapPath("StoreUserInfo.xml"));
      int i = 1;
      Label1.Text = i.ToString();
      DataTable dt = ds.Tables[0];
      for(i=1;i<=dt.Rows.Count;i++)
      {
        if (i <= dt.Rows.Count)
        {
          DataRow dr = dt.Rows[i - 1];
          string Str = dr["name"].ToString();
          if (Str == TextBox5.Text)
          {
            TextBox1.Text = dr["name"].ToString();
            TextBox2.Text = dr["Email"].ToString();
```

```
            TextBox3.Text = dr["Comment"].ToString();
        }
        else
            Label2.Text = "not found";
    }
}
```

في الكود السابق قمنا بتعريف Dataset وهي ds وجعلناها تقرأ من الملف الخاص بنا والذي يحمل الاسم StoreUserInfo.xml من خلال التعليمة

```
ds.ReadXml(Server.MapPath("StoreUserInfo.xml"));
```

حيث أن التابع ReadXML هو أحد توابع dataset الذي مهمته قراءة ملف xml وتخزينه ضمن dataset على هيئة جداول.

ومن ثم قمنا بتعريف DataTable باسم dt والتي سنخزن فيها جدول Dataset السابق

ثم قمنا بوضع حلقة for للدوران والبحث ضمن DataRows الخاصة بالـ DataTable السابقة لايجاد الاسم الذي قمنا بإدخاله ضمن مربع النص .

فوضعنا الشرط if للمقارنة بين قيمة المتحول i وهو العداد وبين عدد اسطر الجدول في dataTable ليكون سببا بإنهاء الحلقة.

الآن نمر على اسطر الجدول سطرا سطرا

```
DataRow dr = dt.Rows[i - 1];
```

ونقوم بتخزين قيمة السطر في متحول نصي Str

```
string Str = dr["name"].ToString();
```

فإذا كانت قيمة النص الموجود في مربع النص مساوية لقيمة المتحول Str فهذا يعني أني وجدت السطر المطلوب عرضه.

ليتم عرض البيانات كما يلي :

```
Label1.Text = dr["name"].ToString();
Label2.Text = dr["Email"].ToString();
Label3.Text = dr["Comment"].ToString();
```

وبنفس الخطوات السابقة يكون لدينا كود VB.NET التالي :

```
Dim ds As DataSet = New DataSet
ds.ReadXml(Server.MapPath("StoreUserInfo.xml"))
Dim i As Integer = 1
Label1.Text = i.ToString
Dim dt As DataTable = ds.Tables(0)
i = 1
Do While (i <= dt.Rows.Count)
   If (i <= dt.Rows.Count) Then
      Dim dr As DataRow = dt.Rows((i - 1))
      Dim Str As String = dr("name").ToString
      If (Str = TextBox5.Text) Then
         TextBox1.Text = dr("name").ToString
         TextBox2.Text = dr("Email").ToString
         TextBox3.Text = dr("Comment").ToString
      Else
         Label2.Text = "not found"
      End If
   End If
   i = (i + 1)
Loop
```

نهاية الدورة

أرجو أن أكون قد وفقت في شرح لغة XML لكم بشكل جيد

زميلكم Tammam