

# البيانات شبه المنظمة

## I.XML

### 1. مقدمة

#### تاريخية

يبدو أن XML يشبه HTML، ولكنه مختلف تماماً. كلاهما خلفاء SGML، نفسها مشتقة من GML. تم تصميم هذا الأخير بواسطة IBM لفصل مظهر المستندات النصية عن محتواها، وخاصة الوثائق الفنية. من السهل تغيير المظهر (التخطيط، الخطوط، الألوان، إلخ) لمثل هذا المستند دون إعادة كتابة سطر واحد. اللغات الأخرى، مثل LaTeX، متشابهة: فنحن نكتب النص دون القلق بشأن التخطيط. وهذا يسمح أيضاً بترجمتها إلى لغات أخرى. وعلى العكس من ذلك، البرمجيات *wysiwyg* مثل تخطيط ومحتوى مزيج Word. حتى مع الأنماط، لا يزال من الصعب تغيير مظهر المستند دون الحاجة إلى إعادة كتابته جزئياً على الأقل.

XML يمكن اعتباره تعميماً لـ SGML و HTML مما يسمح ببناء جميع أنواع المستندات.

#### "(XML لغة الترميز الموسعة) ؟"

إنها لغة تسمح لك بتمثيل المعلومات وتنظيمها باستخدام العلامات. يمكن للجميع تحديد العلامات واستخدامها كما يريدون.

مثال: نص ... <TAG> ... نص ... </TAG> ...

#### لماذا لغة XML؟

يقع تنسيق XML في قلب العديد من العمليات الحالية:

- تبادل البيانات بين الخوادم والعملاء،
  - أدوات البرمجة واللغات،
  - قواعد بيانات XML الأصلية.
- يحدد XML بناء جملة المستند، لكن التطبيقات تحدد العلامات وما تعنيه وما يجب أن تحتوي عليه. توجد أوجهات برمجة التطبيقات بجميع اللغات لقراءة وكتابة مستندات XML.

#### مثال لملف XML

يمثل ملف XML معلومات منظمة (مكتوبة بشكل جيد). يوضح المثال التالي مساراً يتكون من مراحل:

```
<?xml>الإصدار = "1.0" الترميز = "utf-8"?<!-- طريق خيالي
<!--> الطريق
```

```
<خطوة المسافة = "0 كم">رحيل</خطوة> <خطوة المسافة = "13 كم">
انعطف يمينا</خطوة> <خطوة المسافة = "22 كم">وصول</خطوة> </
الطريق>
```

نختار العلامات والصفات كما نرغب. يجب فقط أن تكون متجانسة ومنتظمة بحيث يمكن استخدامها بواسطة البرنامج (المحلل اللغوي).

يمكننا القيام بذلك، لكن مرحباً بالصعوبات في معالجته:

```
<?xml>الإصدار = "1.0" الترميز = "utf-8"?<!-- طريق خيالي
<!--> الطريق
```

```
<خطوة المسافة = "0 كم">رحيل</خطوة> <خطوة>
المسافة 13 كم</المسافة> دور
<خطوة المسافة = "22 كم">معلومات</المعلومات> <خطوة> </الطريق>
```

## مثال لتطبيق XML

### كتنسيق ملف XML

تستخدم العديد من أدوات تكنولوجيا المعلومات تنسيق XML مباشرة لحفظ ملفاتها:

المكتب: يستخدم LibreOffice التنسيق [افتح المستند](#). الرسم المتجه باستخدام Inkscape، التنسيق SVG. المعادلات الرياضية، الشكل ماثمل، ...

### قواعد بيانات XML

قواعد البيانات الأصلية XML: البيانات بتنسيق XML والاستعلامات موجودة بلغة (XQuery) مما يسمح بإنشاء ما يعادل SQL.

### تبادل البيانات بين العملاء والخادم

وما إلى ذلك AJAX و SOAP و XML-RPC و RSS هو التنسيق المستخدم لتمثيل البيانات المتطابقة من العديد من البروتوكولات مثل XML

## 2. هيكل وثيقة XML

يتكون مستند XML من عدة أجزاء:

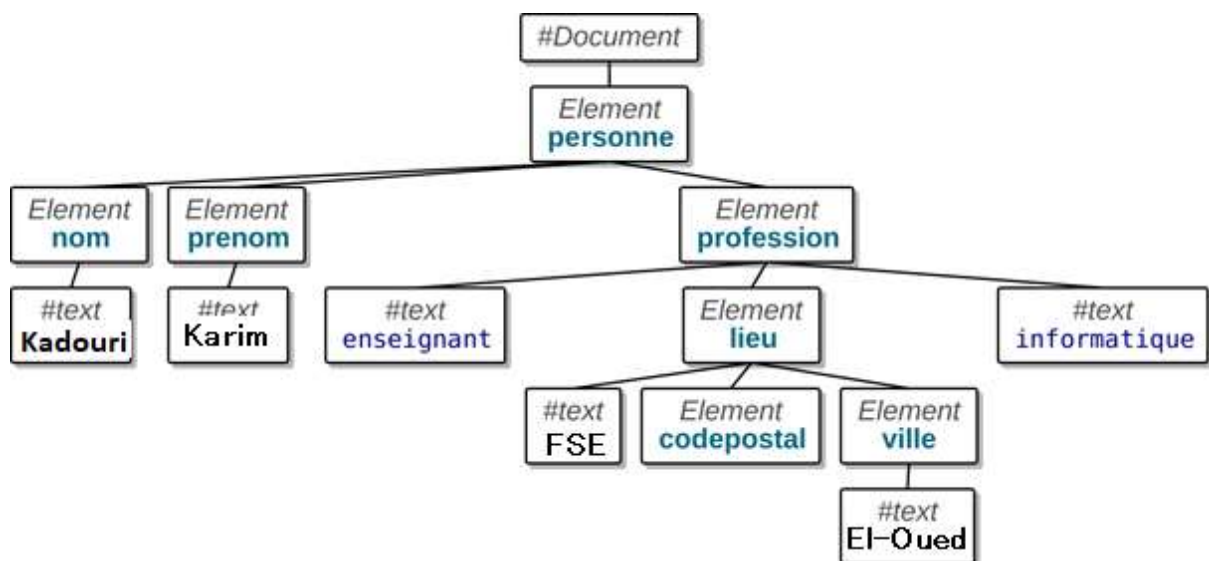
- (أ) رأس المستند الذي يحدد الإصدار والتشفير،
- (ب) قواعد اختيارية تسمح لك بالتحقق مما إذا كانت الوثيقة صالحة "المستند القياسي".
- (ج) شجرة من العناصر تبدأ من عنصر يسمى الجذر.
- العنصر له اسم وسمات ومحتوى
- محتوى العنصر يمكن أن يكون:

\* لا شيء: تمت ملاحظة العنصر الفارغ <الاسم/> أو <صفات الاسم.../>  
\* نص

\*عناصر أخرى (عناصر فرعية).  
 - يتم تحديد العنصر غير الفارغ بعلامة فتح وعلامة إغلاق. \*تمت الإشارة إلى علامة الافتتاح >  
 اسم السمات...< \*تمت الإشارة إلى علامة الإغلاق <name/>

فيما يلي مثال لمستند XML يمثل شخصاً:

```
?xml version="1.0" encoding="utf-8"?<person>
<الاسم>قدوري</الاسم>
<الاسم الأول>كريم</الاسم الأول>
<المهنة>مدرس</المهنة>
<الموقع>
  كلية العلوم الدقيقة
</الرمز البريدي>
<city/>مدينة<العود>
</مكان>
  علوم الكمبيوتر
</المهنة>
</شخص>
```



## برولوج XML

يسمى السطر الأول من مستند XML بمقدمة XML. وهو يحدد إصدار معيار XML المستخدم (1.0 أو 1.1 وهي متشابهة جداً) بالإضافة إلى التشفير:

<?xml الإصدار="1.0" الترميز="utf-8"?>  
 <?xml الإصدار="1.0" ترميز="iso-8859-15"?>

هناك العديد من معايير التشفير:

- أسكي يمثل فقط أول 128 حرفاً من Unicode.
- **إيزو 1-8859** أو Latin-1 يمثل 191 حرفاً من الأبجدية اللاتينية رقم 1 (أوروبا الغربية) باستخدام بايت واحد. الحرفان € و œ ليسا جزءاً منه، على الرغم من وجود æ.
- **إيزو 8859-15** هو المعيار الأكثر ملاءمة للغة الفرنسية. فهو يضيف € و œ ويزيل الأحرف غير المفيدة مثل ð.
- **UTF-8** يمثل الأحرف باستخدام عدد متغير من البايتات، من 1 إلى 4 حسب الحرف.

على سبيل المثال: يتم ترميز A بواسطة 0x41، و é بواسطة 0xA9، و € بواسطة 0xE2، 0x82، 0xAC.

## تعليق XML

فيما يلي مثال لتعليق XML

```
<!-- هذا تعليق -->
```

```
<!--
```

هنا تعليق آخر وهذا --، وهذا خطأ

```
<--
```

القيد الوحيد هو عدم القدرة على استخدام الأحرف -- في التعليق، حتى لو لم يتبعها <.

بعد المقدمة، يمكننا العثور على عدة أجزاء اختيارية محددة بـ <?...?> أو <!...>. هذه هي تعليمات المعالجة والتوجيهات لمحلل XML.

على سبيل المثال:

- **أعريفات نوع الوثيقة** (الذي يسمح لك بالتحقق من صحة محتوى المستند) DTD
- ورقة الأنماط،

**xml** الإصدار = "1.0" الترميز = "utf-8" <? "person.dtd" نظام الشخص DOCTYPE  
!> ورقة الأنماط -> "css" نوع = "نص" href = "person.css" <? < شخص >

```
<...>
</شخص>
```

## 2.1 عناصر XML

يتم تحديد عنصر XML بواسطة:

- علامة افتتاحية <اسم السمة...>
- علامة إغلاق إلزامية </name>.

محتوى العنصر موجود بين العلامتين. هذه هي العناصر الفرعية و/أو النص.

```
<والد>
```

النص 1

```
<الطفل>text2</الطفل>
```

text3

```
</والد>
```

في حالة عدم وجود محتوى للعنصر (عنصر فارغ)، يمكننا تجميع هاتين العلامتين في <...attribute name>/> واحد

يمكن العثور على نفس نوع العنصر عدة مرات مع نفس الأصل، بمحتويات متطابقة أو مختلفة:

<العنصر1>  
 <العنصر2>النص 1</العنصر2> <العنصر  
 2>text2</العنصر2> <العنصر2>النص 1  
 </العنصر2> </العنصر1>

الحرف:يسمح لك بفصل الاسم إلى جزأين: البادئة والاسم المحلي. الأمر كله يسمى اسم مؤهل. على سبيل المثال، iut:departement هو اسم مؤهل يسبقه iut. تتيح لك هذه البادئة تحديد مساحة الاسم.

اسم مؤهل=بادئة:الاسم المحلي

## مساحات الأسماء

تحدد مساحة الاسم مجموعة من الأسماء لتجنب الخلط بين العناصر التي لها نفس الاسم ولكن ليس نفس المعنى. يحدث هذا عندما يقوم مستند XML بتكوين معلومات من مجالات متعددة.

مثال مع الارتباك:

```
<الأثاث معرف="765">
  <table> السعر="€74.99">المنتج</table>
  <table> الحدود="1">
    <tr><td>سم 120</td><td>سم 80</td></tr></table>
    <tr><th>العرض</th><th>الطول</th></tr></table>
```

</الأثاث>

الآن هنا هو المثال السابق مع مساحات الأسماء:

```
<?xml version="1.0" encoding="utf-8"?>
  <الأثاث:الأثاث معرف="765">
    xmlns:html="http://www.w3.org"
    xmlns:furniture="urn:iutlan:furniture"
    <furniture:table Price="€74.99">الماهو جني</furniture:table>
    <html:table>
      <tr>
        <th>العرض</th>
        <th>الطول</th>
      </tr>
      <tr>
        <td>سم 120</td>
        <td>سم 80</td>
      </tr>
    </table>
```

</الأثاث:الأثاث>

عندما يقوم جذر المستند بتعريف سمة URI = "xmlns"، فسيتم وضع جميع علامات المستند بشكل افتراضي في مساحة الاسم هذه، لذلك لا حاجة لوضع بادئة. يمكن أيضاً تطبيق هذا محلياً على العنصر ويتعلق أيضاً بجميع نسله:

```
<الأثاث:الأثاث معرف="765" xmlns="urn:iutlan:furniture">
  <table> السعر="74.99 يورو">الماهو جني</table>
  <table> الحدود="1" xmlns="http://www.w3.org">
    <tr><th>العرض</th><th>الطول</th></tr>
    <tr><td>سم 120</td><td>سم 80</td></tr>
  </table>
</الأثاث>
```

## صفات

السمات تميز العنصر. هذه هي أزواج الاسم = "القيمة" أو الاسم = "القيمة". يتم وضعها في العلامة الافتتاحية.

مثال :

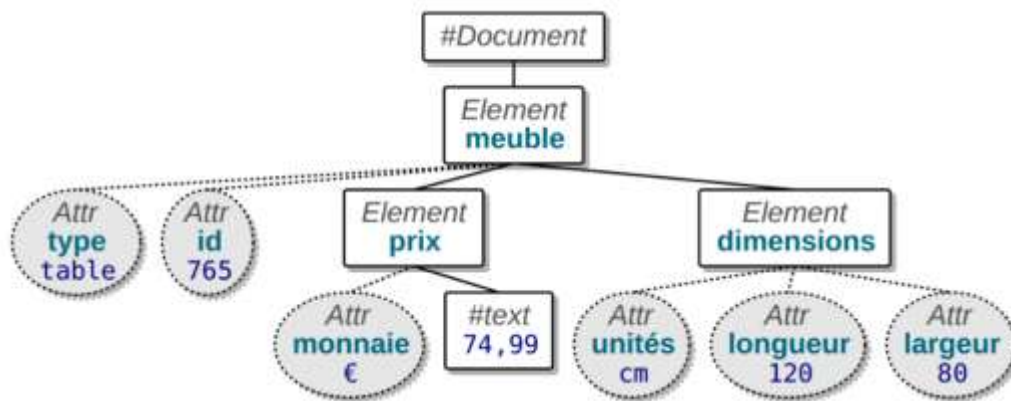
<?xml الإصدار = "1.0" الترميز = "utf-8"?>

<الأثاثالمعرف="765" اكتب ='طاولة'>

<السعرالعملة="74.99"/>السعر<

<الأبعادالوحدات="سم"الطول="120"العرض="80"/>

الوصفاللغة="أون"<طاولة صغيرة لطيفة/>وصف</أثاث>



ملاحظات:

لا يوجد ترتيب بين سمات العنصر. يمكن أن تكون السمة موجودة مرة واحدة فقط.

## 3. جداول النمذجة (الكيانات)

خذبعين الاعتبار جدول السيارة التالي:

سيارة
المعرف:سلسلة العلامة التجارية: ثونغ اللون:ثونغ

فيمايلي تمثيلات XML المحتملة لهذا الجدول:

<pre> &lt;?xml version="1.0" encoding="utf-8" &gt; &lt;سيارات&gt;   &lt;سيارة&gt;     &lt;المعرف&gt;125&lt;/المعرف&gt;     &lt;العلامة التجارية&gt;رينو&lt;/العلامة التجارية&gt;     &lt;اللون&gt;رمادي&lt;/اللون&gt; &lt;/&gt;   &lt;/سيارة&gt;   &lt;سيارة&gt;     &lt;المعرف&gt;982&lt;/المعرف&gt;     &lt;العلامة التجارية&gt;بيجو&lt;/العلامة التجارية&gt;     &lt;اللون&gt;أسود&lt;/اللون&gt; &lt;/&gt; &lt;/سيارات&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="utf-8" &gt; &lt;سيارةالمعرف="125"العلامة التجارية="رينو"اللون="رمادي"&gt; &lt;/سيارةالمعرف="982"العلامة التجارية="بيجو"اللون="أسود"&gt; &lt;/سيارات&gt; </pre>
--	---

## صفات أم عناصر فرعية؟

عند التفكير في اختيار تمثيل المعلومات في شكل سمات أو عناصر فرعية،

ويجب أن تؤخذ هذه المعايير بعين الاعتبار:

- لا يمكن أن تحتوي السمات على عدة قيم، بينما يمكن أن يكون لدينا عدة عناصر فرعية بنفس الاسم ومحتويات مختلفة،
- لا يمكن أن تحتوي السمات على هياكل شجرية، على عكس العناصر،
- السمات ليست قابلة للتوسيع بسهولة، في حين يمكن دائماً إضافة عناصر فرعية جديدة إلى التسلسل الهرمي دون تغيير البرامج الموجودة.
- إن الوصول إلى العناصر الفرعية ليس أكثر تعقيداً من الوصول إلى السمات لأن جميعها عبارة عن عقد في الشجرة الأساسية

## 4 التحقق من صحة وثيقة XML

### لماذا التحقق من الصحة:

تبادل البيانات، تخيل أن الشخص X يريد تبادل البيانات المتعلقة بالطلاب: اللقب والاسم الأول وتاريخ الميلاد والمعدل وما إلى ذلك. يجب أن يتفق كلا المراسلين على تنسيق البيانات؟

وهو شكل أكثر حداثة ولكنه أيضاً أكثر تعقيداً من القواعد النحوية W3C، كونها الشكل الأقدم، موجودة في معظم الأدوات. بعد ذلك يأتي ما نسميه مخططات (تعريف نوع المستند) DTDs

## 4.1 التحقق من صحة DTD

يعد DTD (تعريف نوع المستند) شكلاً قديماً نسبياً من القواعد النحوية لأنه يأتي من عالم SGML (لغة التوصيف المعممة القياسية).

على سبيل المثال، مستند XML مع DTD cours.dtd خارجي، الموجود في نفس الدليل مثل مستند XML الخاص بنا (الوصول النسبي)، يكون في النموذج:

```
<?xml version="1.0"?>
<!DOCTYPE "course.dtd"
  <الدورة>
```

...  
</> بالطبع

في هذا المثال، يقع DTD cours.dtd بالنسبة إلى مستند XML الخاص بنا. تعتبر الكلمة الأساسية SYSTEM مهمة وتشير إلى أن هذا هو DTD فريد بالنسبة لك. البديل هو الكلمة الأساسية العامة.

## تعريف العنصر

```
<!ELEMENT NameElemnt DEF_CONTENT>
```

قديحتوي DEF\_CONTENT على:

- فارغ: العنصر لا يحتوي على محتوى. ولذلك فهو فارغ. ومع ذلك، يمكن أن يكون لها سمات.
- أي: يمكن أن يحتوي العنصر على أي عنصر موجود في DTD.
- PCDATA يحتوي العنصر على نص. يوجد الحرف # لتجنب الغموض مع العلامة ويخبر المحلل اللغوي أنها كلمة رئيسية. يعني: #PCDATA) بيانات الأحرف القابلة للتحليل.

- عنصر بين قوسين مثل (element\_name). يشير اسم العنصر إلى مرجع لعنصر موصوف

في جزء آخر من DTD

- مجموعة من العناصر مفصولة بعوامل، جميعها محاطة بين قوسين. عامل الاختيار، الذي يمثله الحرف "|"، يشير إلى وجود أي من العنصرين (أو مجموعتين من العناصر). عامل التسلسل (أو التسلسل)، الذي يمثله الحرف "،"، يشير إلى أن كلا العنصرين (أو مجموعتي العناصر) يجب أن يكونا موجودين. يمكن استخدام أقواس إضافية لحل الغموض

بعض الأمثلة:

<عنصرالشخص (الاسم\_الأول | اسم العائلة)>



<عنصر اسم\_العائلة (PCDATA#) > !عنصر  
الاسم (PCDATA#) <

وهذا هو شكل النحو:

شخص- الاسم الأول | اسم\_ اسم  
العائلة- نص  
الاسم- النص

وهذا يسمح لنا بوثيقتين XML، وهما:

<شخص>  
</first\_name> <كريم قدوري> /first\_name </person>

أو:  
<شخص>  
<الاسم> <قدوري> /الاسم </شخص>  
حالة أخرى مع عامل التسلسل.

<PCDATA# (اسم العائلة ELEMENT >  
<PCDATA# (الاسم الأول ELEMENT > !ELEMENT > شخص )  
الاسم الأول، اسم العائلة ELEMENT > !ELEMENT

إنه مثل النحو:

الشخص- الاسم الأول الاسم الأخير  
الاسم الأول - النص  
الاسم- النص

هنا، يجد عامل التسلسل الإمكانات في مستند XML واحد صالح:

<شخص>  
<الاسم الأول> <كريم> /الاسم الأول </شخص>  
<الاسم> <قدوري> /الاسم </شخص>

~~<شخص>  
<الاسم> <قدوري> /الاسم الأول </شخص>  
<الاسم الأول> <كريم> /الاسم </شخص>~~

يمكن قياس المحتويات (عنصر أو مجموعة عناصر) بواسطة العوامل \*،  
+ و. ترتبط هذه العوامل بمفهوم العلاقة الأساسية. في حالة عدم وجود عامل تشغيل، يكون  
القياس الكمي 1 (وبالتالي فهو موجود دائماً).  
فيما يلي تفاصيل هؤلاء المشغلين:

\*: من 0 إلى n مرة؛

- +: من 1 إلى n مرة؛
- ؟: 0 أو 1 مرة.

بعض الأمثلة:

<عنصر الخطة (المقدمة؟، الفصل +، الخاتمة؟)>

يحتوي عنصر المخطط التفصيلي على عنصر مقدمة اختياري، يتبعه عنصر فصل واحد على الأقل وينتهي بعنصر خاتمة اختياري أيضاً.

<عنصر الفصل (المؤلف\*، الفقرة+)>

يحتوي عنصر الفصل على 0 إلى n من عناصر المؤلف متبوعة بعنصر فقرة واحد على الأقل.

<عنصر الكتاب (المؤلف؟، الفصل+)>

يحتوي عنصر الكتاب على عنصر واحد على الأقل، ويكون كل عنصر عبارة عن مجموعة من العناصر حيث يكون عنصر المؤلف اختيارياً وعنصر الفصل موجوداً في نسخة واحدة.

## تحديد السمات

يتم تحديد السمات في التعليمات **ATTLIST**. هذا الأخير، كونه مستقلاً عن التعليمات **عنصر**، نحدد مرة أخرى اسم العنصر الذي تنطبق عليه السمة (السمات). يمكننا أن نعتبر أن هناك هذا الشكل النحوي:

```
ELEMENT_NAME ATTLIST>
Attr_def* ... <
```

تم تفصيل Attr\_def على النحو التالي:

اسم نوع الالتزام DEFAULT\_VALUE

يمكن أن يكون TYPE بشكل أساسي:

• بيانات CDATA: نص (بيانات الشخصية) ;

• المعرف: معرف فريد (مزيج من الأرقام والحروف)؛

• IDREF: إشارة إلى معرف

قائمة المراجع إلى المعرفات (مفصلة باللون الأبيض)؛ IDREFS:

• NMToken: كلمة واحدة (لذلك لا توجد خانة فارغة)؛

• NMTOKENS: قائمة الكلمات (مفصلة بفراغ)؛

• تعداد القيم: يتم فصل كل قيمة بالحرف |. لا يتعلق الالتزام بالتعدادات التي تتبعها قيمة افتراضية.

وفي حالات أخرى يتم التعبير عنها على النحو التالي:

• # مطلوب: سمة إلزامية.

• # ضمني: سمة اختيارية.

• # تم الإصلاح: السمة لا تزال موجودة بقيمة. ويمكن استخدام هذا، على سبيل المثال، ل

فرض وجود مساحة الاسم.

موجود للتعداد أو عند كتابة القيمة DEFAULT\_VALUE

#ضمني أو #ثابت. بعض

الأمثلة:

فصل ATTLIST!>

عنوان مطلوب #CDATA

مؤلف بيانات CDATA #ضمنية<

يحتوي عنصر الفصل هنا على سمة عنوان إلزامية وسمة مؤلف اختيارية. قلم رصاص ATTLIST!>

لون(أحمر|أخضر|أزرق)"أزرق"<

يحتوي عنصر القلم الرصاص على سمة لون تكون قيمها جزءاً من المجموعة الأحمر والأخضر والأزرق.

## تعريف الكيانات

يتم الإعلان عن الكيانات بواسطة بيان ENTITY. كما ناقشنا في الفصل السابق، يربط الكيان الاسم بالقيمة. يستخدم هذا الاسم في مستند XML كنموذج لاسم مستعار أو اختصار للقيمة التي تتبع بناء الجملة &name; يمكن أن تكون قيمة الكيان داخلية أو خارجية.

إن بناء جملة الإعلان عن الكيان هو ببساطة ما يلي: <!اسم الكيان

"VALUE"> حالة داخلية.

<!اسم نظام الكيان "unText.txt"> حالة خارجية.

## أمثلة:

### مثال 1

<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE الرحلة [

<خط سير رحلة العنصر (حلقة؟، الخطوة، البديل\*)> <!حلقة العنصر فارغة>

<عنصر الخطوة (PCDATA#)> <عنصر  
البديل ANY> [

<الطريق>

</حلقة>

<خطوة> ابدأ </خطوة>

<خطوة> انعطف يميناً </خطوة> <variant>

</variant> <step> <انعطف يساراً> <step> <خطوة> <step>

</الطريق>

مثال 1

```
<?xml version="1.0" encoding="ISO-8859-1"><
  DOCTYPE
  Brilliant" > !ELEMENT Course )#PCDATA(<
    > !ENTITY Author "Alexandre
    <[
      <الدورة>
      الدورة من إنتاج & مؤلف! /> بالطبع<
```

المثال 2

```
>?xml version="1.0" encoding="utf-8"?<
  DOCTYPE الرحلة
  >خط سير الرحلة DOCTYPE
  step )#PCDATA( !اسم مسار < <!(خطوة+)>
  <ATTLIST CDATA #IMPLIED > !ELEMENT
  < < !ATTLIST المسافة CDATA #REQUIRED >
  <خط سير الرحلة الاسم = "اختبار">
  <step distance="1km"> <انعطف يميناً> /step <
  <step distance="0km"> <خط سير الرحلة>
```

### مثال 3

```

<وكالة ELEMENT (الاسم، المكتب، الفترة، الرحلة*) > ATTLIST!>
    مستوى الوكالة CDATA #ضمني<
        الاسم(PCDATA#) <العنصر
        مكتب(PCDATA#) <العنصر
    > ATTLIST! مكتب office_code معرف#مطلوب<
        <عنصر الفترة(PCDATA#) <
            #IMPLIED مستوى السفر (1 | 2 | 3 | 4 | 5) ATTLIST!> <البلد، المدة،
                الوصف، السعر) travel ELEMENT!>
    مسؤول IDREF #مطلوب
    travel_code معرف#مطلوب<
        <عنصر وصفي CDATA "fr" لغة * > ملاحظة | راجع |
        <وصف ATTLIST PCDATA#>
        <ELEMENT see EMPTY >
        <ELEMENT note) PCDATA#( <
        <PCDATA#( راجع المعرفات الأخرى #المطلوبة <!سعر
            العنصر ATTLIST!>
        <عنصر البلد(PCDATA#) <!مدة
            العنصر(PCDATA#) <

```

## XML مخطط 4.2