د. طه زروقي

البيانات شبه الهيكلة

Semi Structured Data

Données semi-structurées

Contents

1 Lessons summaries — **6 ملخصات الدروس**

1.1 Introduction to XML — مقدمة في XML . . . 7
1.2 DTD (Document Type Definition) — تعريف نوع المستند . . . 12
1.3 XML Schema Definition — تعريف مخطط XML . . . 16
1.4 XSLT — ورقة الأنماط والتحويل . . . 20
1.5 XPath — لغة المسارات . . . 24
1.6 XQuery — لغة الاستعلامات . . . 27
1.7 XML DOM — نموذج كائن المستند . . . 32

2 Practical works — **37 أعمال تطبيقية**

2.1 How to write a report — كيفية إعداد تقرير . . . 38
2.2 Practical works No. 1**[Python]** — العمل التطبيقي 1 . . . 40
2.3 Practical works No. 2**[DTD]** — العمل التطبيقي 2 . . . 46
2.4 Practical works No. 3**[XSLT]** — العمل التطبيقي 3 . . . 51
2.5 Practical works No. 4**[XSD]** — العمل التطبيقي 4 . . . 55
2.6 Practical works No. 5**[XQuery]** — العمل التطبيقي 5 . . . 61
2.7 Practical works No. 6**[XML DOM]** — العمل التطبيقي 6 . . . 70

3 Tests — **76 فحوص**

3.1 Tests topics — مواضيع الفحوص . . . 76
3.2 Test solutions — حلول الفحوص . . . 79

4 Exams — **87 امتحانات**

4.1 Topics — مواضيع . . . 87
4.2 Exam solutions — حلول امتحانات . . . 93

5 Appendices — **113 ملحقات**

5.1 Appendice A: Resources — ملحق أ: موارد . . . 113
5.2 Appendice B: The course curriculum — ملحق ب: منهاج المادة . . . 115
5.3 Glossary — مسرد . . . 117

Bibliography — **119**

عن الكتاب About

كتاب «البيانات شبه المهيكلة» مرجعٌ في تعلم تقنيات XML، يجمع بين الدروس النظرية والأعمال التطبيقية، ويهدف إلى تعليم معالجة البيانات شبه المهيكلة بأسلوب عملي موجّه بأمثلة وتمارين ومشاريع.

صُمم هذا الكتاب ليفيد طلبة علوم الحاسوب في مرحلتي الليسانس والماستر، و التخصصات ذات الصلة بتطوير الويب، وهندسة البرمجيات، وقواعد البيانات، ونظم المعلومات. وهو مرجعٌ للمهندسين، والمطورين، وكل من يرغب في استكشاف تقنيات XML وتطبيقاتها.

تمثل تقنيات XML والبيانات شبه المهيكلة جانبًا مهما في تدريب طلبة علوم الحاسوب (الإعلام الآلي) في العديد من الجامعات، لما لها من دور في تمثيل البيانات وتبادلها بين الأنظمة، وفي تقنيات الويب، وخدماته، وقواعد البيانات، ومعالجة الوثائق الرقمية. وقد روعي في إعداد هذا الكتاب أن يتوافق مع المقررات الجامعية المعتمدة في الجامعات الجزائرية، مع الحفاظ على طابعه العام حتى يكون مرجعًا مناسبًا للطلبة والمهتمين في مختلف المؤسسات التعليمية

يغطي الكتاب أهم تقنيات البيانات شبه المهيكلة، ومن أبرزها:

- لغة XML.
- توصيف بنية المستندات باستخدام DTD وXSD) XML Schema).
- لغة XPath.
- لغة XQuery.
- تحويل مستندات XML باستخدام XSLT.
- معالجة مستندات XML برمجيًا باستخدام DOM ولغة Python.

يعتمد الكتاب منهجية قائمة على التعلم بالممارسة، حيث يتضمن كل فصل عملًا تطبيقيًا موجهًا يشرح المفاهيم الأساسية من خلال التطبيق العملي، يليه عدد من التمارين والواجبات المنزلية المخصصة للتعلم الذاتي، ثم مشاريع تطبيقية لتوظيف المعارف المكتسبة في مسائل واقعية. هذا التدرج يُكسِبُ القارئ مهارات عملية وفهمًا للنظري.

هذا الكتاب ثمرةُ سنوات من تدريس تقنيات XML في مؤسسات التعليم العالي، والإشراف على الأعمال التطبيقية والمشاريع، إلى جانب الاستفادة من الخبرة البحثية في نمذجة المصحف الشريف باستخدام تقنيات XML ضمن رسالة الماجستير.

وقد أُعِدّ الكتاب بصيغة ثنائية اللغة (العربية والإنجليزية)، مع إرفاق مسرد للمصطلحات التقنية، لتسهيل فهم المفاهيم والربط بين المصطلحات العربية ونظيراتها الإنجليزية.

آمل أن يسهم هذا الكتاب في تبسيط تعلم تقنيات البيانات شبه المهيكلة، وأن يكون نافعًا للطلبة والمهتمين. كما أرحب بالملاحظات والاقتراحات التي تسهم في تحسينه مستقبلًا.

المؤلف: د. طه زروقي
taha.zerrouki (at) gmail(dot)com

عن المؤلف

الدكتور طه زروقي، أستاذ بجامعة البويرة في قسم علوم الحاسوب، متخرج من المدرسة الوطنية العليا للإعلام الآلي، مطوّر برمجيات حرة مفتوحة المصدر خاصة باللغة العربية **مهتم ب :**

- المعالجة الآلية للغات الطبيعية
- المصادر المفتوحة

قدّم دروسا في :

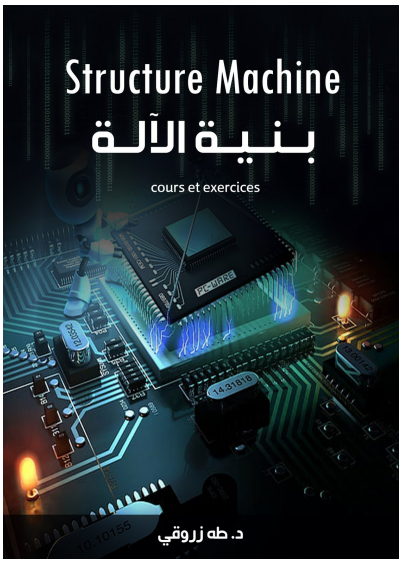
- بنية الآلة ومعمارية الحاسوب
- لغات البرمجة
- تقنيات XML
- تقنيات ومنهجيات البرمجة
- أنماط التصميم Design Pattern
- برامج إدارة المشاريع

كتب للمؤلف :

- دليل المصطلحات لطلبة السنة الأولى الشعب التقنية، إلكتروني، 2012
- كتاب بنية الآلة 1 (عربي/فرنسي) مطبوع، مكتبة عكاشة، 2022
- كتاب بنية الآلة 1 (عربي/فرنسي) إلكتروني، 2021
- كتاب بنية الآلة 2 (عربي/فرنسي) إلكتروني، 2022
- كتاب بنية الآلة 1 (عربي/إنجليزي) إلكتروني، 2023
- كتاب منهجية البرمجة وتقنياتها (عربي/إنجليزي) إلكتروني، 2023

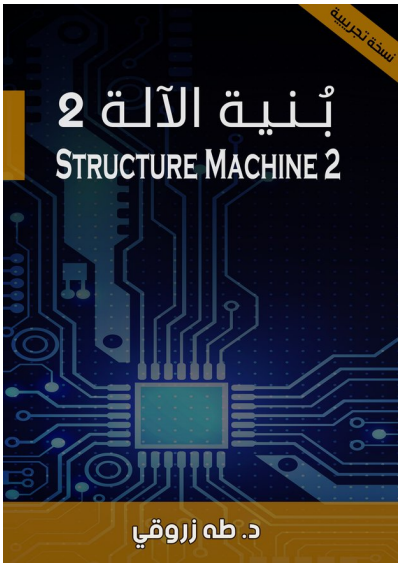
موقع : `http://tahadz.com`
قناة الدروس على يوتيوب
`http://youtube.com/@taha.zerrouki`

كتب للمؤلف

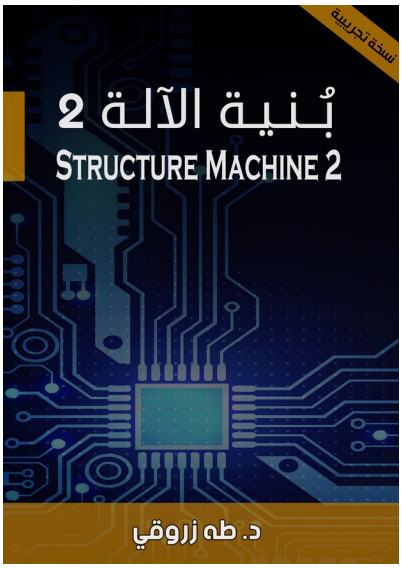


للتحميل

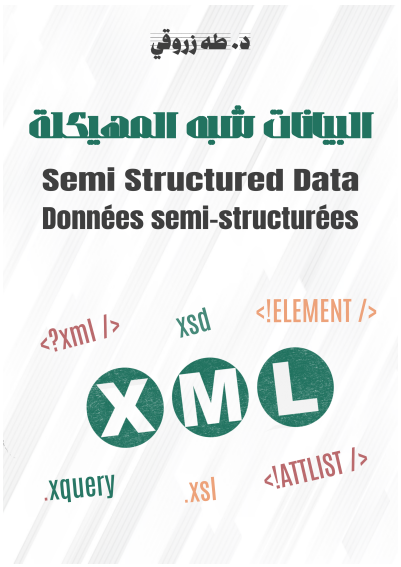
للتحميل



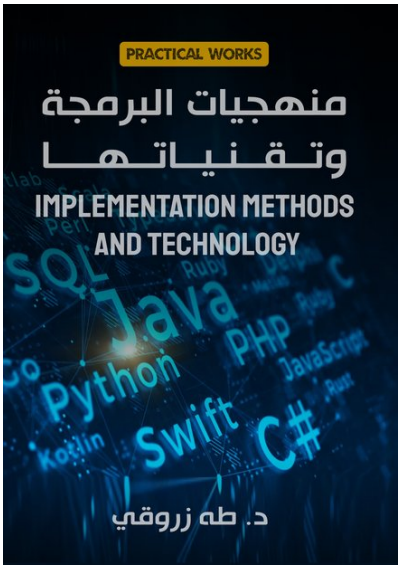
للتحميل



للتحميل



للتحميل



للتحميل

للتحميل

للتحميل

للتحميل

قناة يوتيوب

قائمة التشغيل

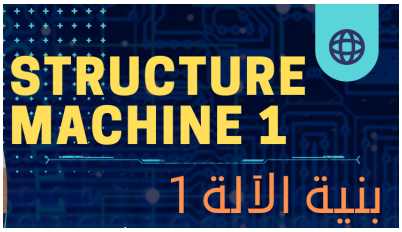
taha.zerrouki@

قائمة التشغيل

قائمة التشغيل

قائمة التشغيل

قائمة التشغيل



قائمة التشغيل

قائمة التشغيل

قائمة التشغيل

قائمة التشغيل

شكر وعرفان

جزيل الشكر للمهندس هيثم بن حليمة لمساهمته في تصميم الغلاف،
والشكر موصول لكل من ساهم من قريب أو بعيد في صياغة هذا الكتاب.

بيان حول الشفافية واستخدام التقنيات الرقمية

نؤكد في هذا العمل أن الأفكار والخطط التدريبية، والتمارين هي نتاج جهد فكري وميداني أصيل للمؤلف.
وتماشياً مع التطور التكنولوجي، استعنت بأدوات الذكاء الاصطناعي والبرمجة للمعالجة القبلية للنصوص بصفتها أدوات مساعدة في الجوانب التقنية الهندسية؛ وشمل ذلك تصفية الأسطر، وتنظيم ملفات النصوص والـ LaTeX، وتوليد الكشافات البرمجية آلياً لتسريع إخراج هذا الكتاب وتنظيمه بأعلى دقة ممكنة، ومراجعة البيانات والتمارين والنصوص.

version 1.0, date: August 3, 2026.

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.

1. Lessons summaries ملخصات الدروس

Contents

1.1 Introduction to XML مقدمة في XML 7
1.2 DTD (Document Type Definition) تعريف نوع المستند 12
1.3 XML Schema Definition تعريف مخطط XML 16
1.4 XSLT ورقة الأنماط والتحويل 20
1.5 XPath لغة المسارات 24
1.6 XQuery لغة الاستعلامات 27
1.7 XML DOM نموذج كائن المستند 32

Introduction to XML

مقدمة في XML

XML, or **eXtensible Markup Language**, is a widely used format for storing and transporting data. It was designed to be both **human-readable** and **machine-readable**, making it a versatile tool for data interchange. XML originated from **SGML** (**Standard Generalized Markup Language**) and was developed to simplify data sharing across different systems and platforms.

XML أو **لغة التوصيف الموسعة (eXtensible Markup Language)**، هي صيغة واسعة الاستخدام لتخزين ونقل البيانات. صُمّمت لتكون قابلة للقراءة **للبشر وللآلات**، مما يجعلها أداة متعددة الاستخدامات لتبادل البيانات. نشأت XML من **لغة التوصيف المعيارية العامة (SGML)** وطُوّرت لتبسيط مشاركة البيانات عبر أنظمة ومنصات مختلفة.

Key Characteristics of XML

- **Self-descriptive:** XML documents describe their own structure and content.
- **Platform-independent:** XML can be used on any operating system or programming language.
- **Extensible:** Users can define their own tags and structures to suit their needs.

مزايا XML

- **وصف ذاتي:** مستندات XML تصف هيكلها ومحتواها بنفسها.
- **مستقلة عن المنصات:** تُستخدم XML على أي نظام تشغيل أو لغة برمجة.
- **قابلة للتوسيع:** يمكن للمستخدمين تعريف وسومهم وهياكلهم الخاصة لتلبية احتياجاتهم.

XML and Related Technologies XML is often used in conjunction with other technologies:

- **XPath:** A language for navigating XML documents and selecting nodes.
- **XSLT:** Used to transform XML documents into other formats, such as HTML or text.
- **XQuery:** A query language for extracting data from XML documents.
- **Web Services:** XML is used in SOAP-based web services for data exchange.

XML والتقنيات المتصلة غالبًا ما تُستخدم XML مع تقنيات أخرى:

- **XPath:** لغة للتنقل في مستندات XML وتحديد العقد.
- **XSLT:** تُستخدم لتحويل مستندات XML إلى تنسيقات أخرى، مثل HTML أو النص.
- **XQuery:** لغة استعلام لاستخراج البيانات من مستندات XML.
- **خدمات الويب:** تُستخدم XML في خدمات الويب القائمة على SOAP لتبادل البيانات.

Advantages

- Human-readable and self-descriptive.
- Platform-independent and widely supported.
- Extensible and flexible for various use cases.

المزايا

- قابلة للقراءة من البشر ووصف ذاتي.
- مستقلة عن المنصة ومدعومة على نطاق واسع.
- قابلة للتوسيع ومرنة للعديد من حالات الاستخدام.

Disadvantages

- Verbose and can be inefficient for large datasets.
- More complex to parse compared to JSON.

العيوب

- مطولة وقد تكون غير فعالة لمجموعات البيانات الكبيرة.
- معقدة في التحليل مقارنة بـ JSON.

Applications of XML

- **Data Interchange:** Used in APIs, configuration files, and data storage.
- **Industry Standards:** Examples include RSS feeds, Microsoft Office files (`.docx`, `.xlsx`), and SVG graphics.
- **Web Development:** XML was widely used in early web development for structuring data.

Tools for Working with XML

- **Editors:** Notepad++, XMLSpy, Visual Studio Code.
- **Parsers:** DOM (Document Object Model) and SAX (Simple API for XML).
- **Libraries:** Python's `xml.etree`, Java's JAXB, and others.

تطبيقات XML

- **تبادل البيانات:** تُستخدم في واجهات برمجة التطبيقات (APIs)، وملفات التكوين، وتخزين البيانات.
- **معايير الصناعة:** أمثلة تشمل خلاصات RSS، وملفات Microsoft Office (`.docx`، `.xlsx`)، ورسومات SVG.
- **تطوير الويب:** تم استخدام XML على نطاق واسع في تطوير الويب المبكر لتنظيم البيانات.

أدوات للعمل مع XML

- **المحررات:** XMLSpy, Visual Studio Code.
- **المحللات:** DOM (نموذج كائن المستند) وSAX (واجهة برمجة التطبيقات البسيطة لـ XML).
- **المكتبات:** `xml.etree` في Python، JAXB في Java، وغيرها.

XML Syntax and Structure

XML documents are structured using a set of rules that ensure consistency and clarity.

Basic Syntax Rules

- XML documents consist of **elements**, which are defined by **tags**.
- Tags are enclosed in angle brackets (`<tag>`), and every opening tag must have a corresponding closing tag (`</tag>`).
- Elements can be nested within other elements but must be properly closed in the correct order.
- XML is **case-sensitive**, so `<Name>` and `<name>` are considered different tags.

XML Declaration Every XML document typically starts with a declaration:

بنية XML وصياغتها

تتم تنظيم مستندات XML باستخدام مجموعة من القواعد التي تضمن الاتساق والوضوح.

قواعد بناء الجملة الأساسية

- تتكون مستندات XML من **عناصر**، يتم تعريفها بواسطة **وسوم**.
- تُحاط الوسوم بأقواس حادة (علامة أكبر وعلامة أصغر) (`<tag>`)، ويجب أن يكون لكل عنصر وسم فتح ووسم إغلاق مقابل (`<tag/>`).
- قد تتداخل العناصر فيما بينها، ولكن يجب إغلاقها بالترتيب الصحيح.
- XML **حساسة لحالة الأحرف**، لذا `<Name>` و `<name>` تعتبران علامتين مختلفتين.

تعريف XML يبدأ كل مستند XML عادةً بتصريح:

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

This specifies the XML version and character encoding.

يحدد هذا التصريح إصدار XML وترميز الأحرف.

Root Element Every XML document must have a single **root element** that contains all other elements. For example:

العنصر الجذري يجب أن يحتوي كل مستند XML على **عنصر جذري** واحد يحتوي على جميع العناصر الأخرى. على سبيل المثال:

```
<root>
  <child>Content</child>
</root>
```

Well-Formed XML A well-formed XML document adheres to all syntax rules, such as proper nesting and closing of tags.

توافق XML مع القواعد يجب أن يتوافق مستند XML بجميع قواعد بناء الجملة، مثل التداخل الصحيح وإغلاق الوسوم.

XML Components

مكونات XML

XML documents are made up of several key components:

تتكون مستندات XML من عدة مكونات رئيسية هي:

Elements are the building blocks of XML. For example:

العناصر هي اللبنات الأساسية لـ XML. على سبيل المثال:

```
<book>
  <title>Introduction to XML</title>
  <author>Ahmed Boualem</author>
</book>
```

Attributes provide additional information about elements. For example:

السمات توفر معلومات إضافية عن العناصر. على سبيل المثال:

```
<book category="education">
  <title>Introduction to XML</title>
</book>
```

Entities Used to represent special characters. For example:

الكيانات تُستخدم لتمثيل الأحرف الخاصة. على سبيل المثال:

```
<message>This is a &lt;tag&gt;</message>
```

Comments are added using `<!-- -->` and are ignored by XML parsers:

التعليقات تُضاف التعليقات باستخدام `<!-- -->` وتتجاهلها محللات XML، وتوضع لتجاهل أقسام ما أو إضافة توضيحات:

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

CDATA Sections are used to include text that should not be parsed by the XML parser:

أقسام CDATA تُستخدم لتضمين نص يجب أن لا يعالجه محلل XML بل يتعامل معه كما هو:

```
<![CDATA[<message>This is unparsed text</message>]]>
```

XML Namespaces

مساحات الأسماء في XML

Namespaces are used to avoid element name conflicts, especially when combining XML documents from different sources.
Namespaces are declared using the `xmlns` attribute:

تُستخدم مساحات الأسماء لتجنب تعارض أسماء العناصر، خاصة عند دمج مستندات XML من مصادر مختلفة.
تُعرّف مساحات الأسماء باستخدام السمة `xmlns`:

```xml
<root xmlns:edu="http://example.com/education">
  <edu:book>Introduction to XML</edu:book>
</root>
```

They allow elements with the same name to coexist by assigning them to different namespaces.

تسمح للعناصر التي لها نفس الاسم بالتواجد في نفس الملف على الرغم من انتمائها إلى مساحات أسماء مختلفة.

XML Validation

التحقق من صحة XML

XML documents can be validated to ensure they follow a specific structure.

يمكن التحقق من صحة مستندات XML للتأكد من اتباعها لهيكل محدد.

Document Type Definition (DTD) defines the structure of an XML document:

تعريف نوع المستند (DTD) يحدد هيكل مستند XML:

```xml
<!DOCTYPE note [
  <!ELEMENT note (to, from, body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
```

XML Schema (XSD) is a more powerful alternative to DTDs, providing data type support and namespace integration:

مخطط XML (XSD) بديل أكثر قوة لـ DTD، يوفر دعمًا لأنواع البيانات ودمج مساحات الأسماء:

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Conclusion

XML remains a powerful and flexible format for representing and exchanging data. It continues to be essential across various industries and applications. Gaining a solid understanding of its structure, syntax, and associated technologies is key to using XML effectively.

خاتمة

لا يزال XML تنسيقًا قويًا مرنًا لتمثيل البيانات وتبادلها، ويلعب دورًا أساسيًا في العديد من الصناعات والتطبيقات. إن فهم بنية XML، وبنيته النحوية، والتقنيات المرتبطة به، أمر ضروري لاستخدامه بشكل فعّال.

DTD (Document Type Definition) تعريف نوع المستند

Introduction

A **Document Type Definition** (**DTD**) is a set of rules that defines the structure, elements, and attributes of an XML document. It serves as a schema or blueprint for validating the correctness of an XML document. DTDs ensure that XML documents adhere to a specific structure and format, making them consistent and predictable.

Key Purposes of DTD:

- Define the **structure** of an XML document.
- Specify the **elements**, **attributes**, and their relationships.
- Enable **validation** of XML documents to ensure they conform to the defined rules.
- **Advantages:**
  - Simple and easy to learn.
  - Provides a basic structure for XML validation.
- **Disadvantages:**
  - Limited data typing (e.g., no support for numbers, dates).
  - No support for namespaces.

مقدمة

يعد **تعريف نوع المستند (DTD)** مجموعة من القواعد التي تحدد بنية مستند XML وعناصره وسماته.

يعمل عمل المخطط أو النموذج للتحقق من صحة مستند XML. تضمن DTDs التزام مستندات XML ببنية وتنسيق معين، مما يجعلها متسقة ويمكن التنبؤ بها.

الأغراض الرئيسية لـ DTD:

- تحديد **بنية** مستند XML.
- تحديد **العناصر والسمات** وعلاقاتها.
- تمكين **التحقق** من مستندات XML لضمان امتثالها للقواعد المحددة.
- **المزايا:**
  - بسيطة وسهلة التعلم.
  - توفر بنية أساسية للتحقق من صحة XML.
- **العيوب:**
  - دعم محدود لأنواع البيانات (مثل عدم دعم الأرقام والتواريخ).
  - لا تدعم مساحات الأسماء (Namespaces).

Types of DTD

DTDs can be defined in two ways:

Internal DTD: The DTD is embedded within the XML document itself.

أنواع DTD

تُعرّف DTD بطريقتين:

DTD داخلي: تُوضع DTD داخل مستند XML نفسه.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note [
  <!ELEMENT note (to, from, body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
  <to>Ahmed</to>
  <from>Boualem</from>
```

```
  <body>Hello, this is a note!</body>
</note>
```

External DTD: The DTD is stored in a separate file and linked to the XML document.

DTD خارجي: توضع DTD في ملف منفصل وتُربط بمستند XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Ahmed</to>
  <from>Boualem</from>
  <body>Hello, this is a note!</body>
</note>
```

The `note.dtd` file would contain:

ملف `note.dtd` يحوي:

```
<!ELEMENT note (to, from, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

DTD Syntax and Components

بنية DTD ومكوناته

DTDs use a specific syntax to define the structure of an XML document. The main components include:
Element Declarations: Define the elements and their content models.

تستخدم DTD صيغًا محددة لتعريف بنية مستند XML. من بين مكوناتها الرئيسية:
تصريحات العناصر: تحدد العناصر ونوع محتواها.

```
<!ELEMENT element_name (content_model)>
```

Example

مثال:

```
<!ELEMENT note (to, from, body)>
```

Attribute Declarations: Define the attributes for elements and their types.

تصريحات السمات: تحدد سمات العناصر وأنواعها.

```
<!ATTLIST element_name attribute_name attribute_type default_value>
```

Example

مثال:

```
<!ATTLIST note id CDATA #REQUIRED>
```

Entity Declarations: Define reusable pieces of text or special characters.

تصريحات الكيانات: تُعرّف أجزاء قابلة لإعادة الاستخدام من النصوص أو الأحرف الخاصة.

```
<!ENTITY entity_name "entity_value">
```

Example

مثال:

```
<!ENTITY company "Hamoud Boualem">
```

Content Models in DTD

أنواع المحتوى في DTD

DTDs allow you to specify the content model for elements, which defines what an element can contain. Common content models include:
#PCDATA: The element contains parsed character data (text).

تتيح DTD تحديد نوع محتوى العناصر، أي ما سيحتويه العنصر. تشمل أنواع المحتوى الشائعة:
#PCDATA: يحتوي العنصر على بيانات نصية قابلة للتحليل.

Example: مثال

```
<!ELEMENT title (#PCDATA)>
```

Child Elements: The element contains other elements.

العناصر الفرعية: يحتوي العنصر على عناصر أخرى.

Example: مثال

```
<!ELEMENT book (title, author)>
```

Mixed Content: The element contains both text and child elements.

المحتوى المختلط: يحتوي العنصر على نصوص وعناصر فرعية.

Example: مثال

```
<!ELEMENT paragraph (#PCDATA | em | strong)*>
```

Occurrence Indicators: Define how many times an element can appear.

مؤشرات التكرار: تحدد عدد مرات تكرار العنصر في العنصر الأصلي.

- * (zero or more)
- + (one or more)
- ? (zero or one)

- * (صفر أو أكثر)
- + (واحد أو أكثر)
- ? (صفر أو واحد)

Example: مثال

```
<!ELEMENT library (book+)>
```

Attribute Types in DTD

أنواع السمات

Attributes in DTDs can have different types, including:

للسمات في DTD أنواع مختلفة، من بينها:

CDATA: Character data (text).

CDATA: بيانات نصية.

Example: مثال

```
<!ATTLIST book id CDATA #REQUIRED>
```

Enumerated: A list of possible values.

قائمة قيم: قائمة بالقيم الممكنة.

Example: مثال

```
<!ATTLIST book category (fiction|nonfiction) #REQUIRED>
```

ID and IDREF: `ID` uniquely identifies an element, while `IDREF` references an `ID`.

ID و IDREF: مُعرّفات `ID` تُحدد عنصرًا فريدًا، بينما `IDREF` مُؤشر إلى ID.

Example: مثال

```
<!ATTLIST book id ID #REQUIRED>
<!ATTLIST author book_id IDREF #REQUIRED>
```

Default Values: Specify a default value for an attribute.

Example:

القيم الافتراضية: تحديد قيمة افتراضية للسمات.
مثال

```
<!ATTLIST book language CDATA "Arabic">
```

Entities in DTD

الكيانات

Entities are used to define reusable pieces of text or special characters.
Internal Entities: Defined within the DTD.

تُستخدم الكيانات لتعريف حروف خاصة أو أجزاء قابلة لإعادة الاستخدام من النصوص.
الكيانات الداخلية: تُعرّف داخل DTD.
مثال

Example:

```
<!ENTITY copyright "© 2024 Ben Badis University">
```

External Entities: Reference external files or resources.

الكيانات الخارجية: تشير إلى ملفات أو موارد خارجية.
مثال

Example:

```
<!ENTITY logo SYSTEM "logo.png">
```

Predefined Entities: Represent special characters, such as:

- `<` for <
- `>` for >
- `&` for &

الكيانات المعرفة مسبقًا: تمثل الأحرف الخاصة، مثل:

- `<` لـ <
- `>` لـ >
- `&` لـ &

Conclusion

خاتمة

DTD is a foundational technology for defining the structure and constraints of XML documents. While it has limitations compared to modern alternatives like XML Schema (XSD), it remains a valuable tool for basic validation and structuring of XML data. Understanding DTD is essential for working with XML in legacy systems and simpler use cases.

تُعدّ DTD تقنية أساسية لتعريف بنية XML وقيودها. وعلى الرغم من محدوديتها مقارنةً بالبدائل الحديثة مثل XML Schema (XSD) ، فإنها لا تزال أداةً قيمة للتحقق من صحة XML في الأنظمة القديمة والحالات البسيطة.

XML Schema Definition

تعريف مخطط XML

Introduction

مقدمة

XML Schema Definition (**XSD**) is a language used to define the structure, content, and data types of XML documents. Unlike DTD, XSD is written in XML syntax and provides more advanced features, such as support for data types, namespaces, and complex structures. XSD is widely used for validating XML documents and ensuring they conform to a specific schema.

Key Purposes of XSD:

- Define the **structure** and **content** of XML documents.
- Specify **data types** for elements and attributes.
- Enable **validation** of XML documents to ensure they adhere to the defined schema.

Advantages of XSD over DTD

- **Data Typing:** XSD supports a wide range of data types (e.g., strings, numbers, dates).
- **Namespace Support:** XSD fully supports XML namespaces, allowing for better modularity and reuse.
- **Extensibility:** XSD allows for the creation of complex types and inheritance.
- **XML Syntax:** XSD is written in XML, making it easier to learn and use for those familiar with XML.

Applications of XSD

- **Data Validation:** Ensuring XML documents conform to a specific structure and data types.
- **Web Services:** Defining the structure of SOAP messages in web services.
- **Configuration Files:** Validating configuration files in software applications.

تعريف مخطط XML (XSD) هو لغة تُستخدم لتعريف مستندات XML من حيث الهيكل والمحتوى وأنواع البيانات. على عكس DTD، يُكتب XSD بتنسيق XML ويوفر ميزات متقدمة، مثل دعم أنواع البيانات والمساحات الاسمية والهياكل المعقدة. يُستخدم XSD على نطاق واسع للتحقق من صحة مستندات XML وضمان توافقها مع مخططٍ معين.

الأغراض الرئيسية لـ XSD:

- تعريف **الهيكل** و **المحتوى** لمستندات XML.
- تحديد **أنواع البيانات** للعناصر والسمات.
- **التحقق من صحة** بنية مستندات XML لضمان تطابقها مع المخطط المحدد.

مزايا XSD على DTD

- **أنواع البيانات:** يدعم XSD مجموعة واسعة من أنواع البيانات (مثل النصوص، الأرقام، التواريخ).
- **دعم المساحات الاسمية:** يدعم XSD بشكل كامل المساحات الاسمية لـ XML، مما يتيح مرونة وإعادة استخدام أفضل.
- **التوسيع:** يسمح XSD بإنشاء أنواع معقدة وتوريثها.
- **ترميز XML:** يتم كتابة XSD بتنسيق XML، مما يسهل تعلمه واستخدامه لمن يجيدون XML.

تطبيقات XSD

- **التحقق من البيانات:** التأكد من توافق مستندات XML مع الهيكل المحدد وأنواع البيانات.
- **خدمات الويب:** تعريف هيكل رسائل SOAP في خدمات الويب.
- **ملفات الإعدادات:** التحقق من صحة ملفات الإعدادات في التطبيقات البرمجية.

XSD Document Structure

هيكلية مستند XSD

An XSD document is an XML file that defines the schema for an XML document. It typically consists of the following components:

مستند XSD هو ملف XML يُعرف المخطط لمستند XML. يتكون عادة من المكونات التالية:

Schema Element: The root element of an XSD document.

عنصر المخطط: العنصر الجذري لمستند XSD.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Schema content goes here -->
</xs:schema>
```

Element Declarations: Define the elements in the XML document.

تصريحات العناصر: تعريف العناصر في مستند XML.

```
<xs:element name="book" type="bookType"/>
```

Type Definitions: Define custom data types for elements and attributes.

تعريف الأنواع: تحديد أنواع البيانات المخصصة للعناصر والسمات.

```
<xs:complexType name="bookType">
  <xs:sequence>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Attribute Declarations: Define attributes for elements.

تصريحات السمات: تعريف السمات للعناصر.

```
<xs:attribute name="id" type="xs:ID"/>
```

XSD Data Types

أنواع البيانات

XSD provides a wide range of built-in data types, including:

يوفر XSD أنواع مدمجة للبيانات، من بينها ذلك:

Simple Types: Basic data types like `xs:string`, `xs:integer`, `xs:decimal`, `xs:date`, etc.

الأنواع البسيطة: أنواع البيانات الأساسية مثل `xs:string`، `xs:integer`، `xs:decimal`، `xs:date`، إلخ.

```
<xs:element name="price" type="xs:decimal"/>
```

Complex Types: Define elements that contain other elements or attributes.

الأنواع المعقدة: تعريف العناصر التي تحتوي على عناصر أخرى أو سمات.

```
<xs:complexType name="bookType">
  <xs:sequence>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Derived Types: Create new types by restricting or extending existing types.

الأنواع المشتقة: إنشاء أنواع جديدة بتقييد الأنواع الموجودة أو توسيعها.

```
<xs:simpleType name="gradeType">
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="20"/>
  </xs:restriction>
</xs:simpleType>
```

Defining Elements and Attributes تعريف العناصر والسمات

Elements: Declared using the `<xs:element>` tag.

العناصر: تُعرّف باستخدام علامة `<xs:element>`.

```
<xs:element name="title" type="xs:string"/>
```

Attributes: Declared using the `<xs:attribute>` tag.

السمات: تُعرّف باستخدام علامة `<xs:attribute>`.

```
<xs:attribute name="id" type="xs:ID"/>
```

Complex Elements: Elements that contain other elements or attributes.

العناصر المعقدة: العناصر التي تحتوي على عناصر أو سمات أخرى.

```
<xs:complexType name="bookType">
  <xs:sequence>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
```

XSD Constraints and Facets القيود والفئات

XSD allows you to define constraints on elements and attributes using **facets**. Common facets include:

يسمح XSD بتعريف القيود على العناصر والسمات باستخدام **الفئات**. تشمل الفئات الشائعة:

Length Constraints: `minLength`, `maxLength`, `length`.

قيود الطول: `minLength` أدنى طول، `maxLength` أقصى طول، `length` الطول.

```
<xs:simpleType name="passwordType">
  <xs:restriction base="xs:string">
    <xs:minLength value="8"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
```

Value Constraints: `minInclusive`, `maxInclusive`, `minExclusive`, `maxExclusive`.

قيود القيمة: `minInclusive` أدنى قيمة تنتمي، `maxInclusive` أقصى قيمة تنتمي، `minExclusive` أدنى قيمة غير منتمية، `maxExclusive` أقصى قيمة غير منتمية.

```
<xs:simpleType name="gradeType">
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="20"/>
  </xs:restriction>
</xs:simpleType>
```

Pattern Constraints: Define a regular expression pattern.

قيود النمط: تعريف نمط تعبير منتظم.

```
<xs:simpleType name="AlgerianPhoneType">
  <xs:restriction base="xs:string">
    <xs:pattern value="0[567]\d{2}-\d{2}-\d{2}-\d{2}"/>
  </xs:restriction>
</xs:simpleType>
```

Namespaces in XSD

مساحات الأسماء

XSD fully supports XML namespaces, allowing you to define and use namespaces in your schema.

يدعم XSD المساحات الاسمية لـ XML دعمًا كاملًا، مما يتيح لك تعريف واستخدام المساحات الاسمية في المخطط.

Target Namespace: Specifies the namespace for the elements defined in the schema.

مساحة الاسم المستهدف: تحدد المساحة الاسمية للعناصر المعرفة في المخطط.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://example.com/books"
           xmlns="http://example.com/books">
  <!-- Schema content goes here -->
</xs:schema>
```

Qualified and Unqualified Elements: Elements and attributes can be qualified (prefixed with a namespace) or unqualified.

العناصر المؤهلة وغير المؤهلة: إذا كانت العناصر مرفقة بمساحة اسمية، تسمى مؤهلة، وإلا تعدّ غير مؤهلة.

Conclusion

خاتمة

XSD is a powerful and versatile tool for defining the structure and constraints of XML documents. Its support for data types, namespaces, and complex structures makes it a preferred choice for modern XML validation.

لغة XSD هي أداة قوية متعددة الاستخدامات لتعريف هيكل مستندات XML وقيودها. إن دعمها لأنواع البيانات والمساحات الاسمية والهياكل المعقدة يجعلها الخيار المفضل للتحقق من صحة XML الحديثة.

XSLT

ورقة الأنماط والتحويل

XSLT (**Extensible Stylesheet Language Transformations**) is a language designed for transforming XML documents into other formats, such as HTML, plain text, or other XML structures. It is part of the **XSL** (**Extensible Stylesheet Language**) family, which also includes **XPath** for navigating XML documents and **XSL-FO** for formatting XML documents.

Key Purposes of XSLT:

- Transform XML data into different formats for presentation or further processing.
- Enable the separation of data (XML) and presentation (HTML, PDF, etc.).
- Facilitate the reuse and repurposing of XML data.

Real-World Applications of XSLT

- **Web Development:** Transforming XML data into HTML for web pages.
- **Data Integration:** Converting XML data from one format to another for integration with different systems.
- **Document Generation:** Generating reports, invoices, or other documents from XML data.

Advantages and Disadvantages of XSLT
Advantages:

- Powerful and flexible for transforming XML data.
- Supports complex transformations and conditional logic.
- Separates data (XML) from presentation (HTML, etc.).

Disadvantages:

- Can be complex to learn and use for beginners.
- Performance may be slower for large XML documents.

XSLT (**تحويلات لغة الأنماط الموسعة**) لغة مصممة لتحويل مستندات XML إلى تنسيقات أخرى، مثل HTML أو النص العادي أو هياكل XML أخرى. وهي جزء من عائلة **XSL** (**لغة الأنماط الموسعة**)، التي تشمل أيضًا **XPath** للتنقل داخل مستندات XML و**XSL-FO** لتنسيق مستندات XML.

الميزات الأساسية:

- تحويل بيانات XML إلى تنسيقات مختلفة للعرض أو المعالجة اللاحقة.
- تمكين الفصل بين البيانات (XML) والعرض (HTML، PDF، إلخ).
- تسهيل إعادة استخدام وإعادة توظيف بيانات XML.

تطبيقات XSLT في العالم الحقيقي

- **تطوير الويب:** تحويل بيانات XML إلى HTML لصفحات الويب.
- **دمج البيانات:** تحويل بيانات XML من تنسيق إلى آخر لدمجها مع أنظمة مختلفة.
- **إنشاء المستندات:** إنشاء تقارير أو فواتير أو مستندات أخرى من بيانات XML.

مزايا وعيوب XSLT
المزايا:

- قوية مرنة في تحويل بيانات XML.
- تدعم التحويلات المعقدة والمنطق الشرطي.
- تفصل البيانات (XML) عن العرض (HTML، إلخ).

العيوب:

- معقدة للتعلم والاستخدام للمبتدئين.
- قد يكون الأداء بطيئًا عند التعامل مع مستندات XML كبيرة.

How XSLT Works

XSLT works by applying a set of rules defined in an **XSLT stylesheet** to an XML document. The XSLT processor reads the XML document and the stylesheet, then produces the output document based on the rules specified.
Key Components:

- **XML Document:** The source data to be transformed.
- **XSLT Stylesheet:** Contains the transformation rules.
- **XSLT Processor:** Software that performs the transformation (e.g., browsers, XSLT libraries).

كيفية عمل XSLT

تعمل XSLT بتطبيق قواعد معرّفة في **ورقة أنماط XSLT** على مستند XML. يقرأ معالج XSLT مستند XML وورقة الأنماط، ثم يُنتج مستندًا جديدًا بناءً على القواعد المحددة.
المكونات الأساسية:

- **مستند XML:** البيانات المصدر التي ستحوّل.
- **ورقة أنماط XSLT:** تتضمن قواعد التحويل.
- **معالج XSLT:** البرنامج الذي يُنفّذ التحويل (مثل المتصفحات أو مكتبات XSLT).

XSLT Syntax and Structure

An XSLT stylesheet is an XML document that follows a specific structure. It typically includes the following components:
Root Element: The root element of an XSLT stylesheet is `<xsl:stylesheet>` or `<xsl:transform>`.

Example:

```
<xsl:stylesheet version="1.0"
            xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <!-- Transformation rules go here -->
</xsl:stylesheet>
```

Templates: Templates define how specific elements in the XML document should be transformed.

Example:

```
<xsl:template match="book">
  <h2><xsl:value-of select="title"/></h2>
  <p>Author: <xsl:value-of select="author"/></p>
</xsl:template>
```

XPath Expressions: XPath is used to navigate and select elements in the XML document.

Example:

```
<xsl:value-of select="book/title"/>
```

بناء وصياغة XSLT

ورقة أنماط XSLT هي مستند XML لها هيكل معيّن. يتضمن عادةً العناصر التالية:
العنصر الجذري: العنصر الجذري في ورقة أنماط XSLT هو `<xsl:stylesheet>` أو `<xsl:transform>`.
مثال

القوالب: القوالب تحدد كيفية تحويل عناصر معينة في مستند XML.
مثال

تعبيرات XPath: يُستخدم XPath للتنقل وتحديد العناصر داخل مستند XML.
مثال

Key XSLT Elements

XSLT provides a variety of elements for defining transformations. Some of the most commonly used elements include: **`<xsl:template>`:** Defines a template for matching specific elements in the XML document.

Example:

العناصر الأساسية

توفر XSLT عناصر متنوعة لتعريف التحويلات. وهذه أكثرها استخدامًا:
عنصر **`<xsl:template>`:** يحدد قالبًا لاختيار عناصر معينة في مستند XML.
مثال

```
<xsl:template match="book">
  <!-- Transformation rules for <book> elements -->
</xsl:template>
```

<xsl:value-of>: Extracts the value of an element or attribute.

عنصر **<xsl:value-of>:** لاستخراج قيمة عنصر أو خاصية.

Example: **مثال**

```
<xsl:value-of select="title"/>
```

<xsl:for-each>: Iterates over a set of nodes and applies templates to each one.

عنصر **<xsl:for-each>:** لاستعراض مجموعة من العقد وتطبيق القوالب على كل منها.

Example: **مثال**

```
<xsl:for-each select="books/book">
  <li><xsl:value-of select="title"/></li>
</xsl:for-each>
```

<xsl:if>: Applies a transformation only if a specified condition is true.

عنصر **<xsl:if>:** يطبق التحويل فقط إذا تحقق شرط محدد.

Example: **مثال**

```
<xsl:if test="price > 20">
  <p>This book is expensive.</p>
</xsl:if>
```

<xsl:choose>: Provides conditional logic with multiple branches (similar to `if-else` statements).

عنصر **<xsl:choose>:** يوفر منطقًا شرطيًا للاختيار من بين فروع متعددة (مشابه لتعابير if-else).

Example: **مثال**

```
<xsl:choose>
  <xsl:when test="price > 20">
    <p>Expensive</p>
  </xsl:when>
  <xsl:otherwise>
    <p>Affordable</p>
  </xsl:otherwise>
</xsl:choose>
```

<xsl:apply-templates>: Applies templates to child elements.

<xsl:apply-templates>: يطبق القوالب على العناصر الفرعية.

Example: **مثال**

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

XSLT and XPath

XSLT relies heavily on **XPath** for selecting and navigating elements in the XML document. XPath expressions are used in various XSLT elements, such as `<xsl:value-of>`, `<xsl:for-each>`, and `<xsl:if>`.
Common XPath Expressions:

- `book/title`: Selects the `<title>` element within a `<book>` element.
- `@id`: Selects the `id` attribute of an element.
- `*`: Matches any element.

الأنماط والمسارات

تعتمد XSLT اعتمادًا كبيرًا على مسارات **XPath** لاختيار وتصفح العناصر في مستند XML.
تُستخدم تعبيرات XPath في العديد من عناصر XSLT، مثل `<xsl:value-of>`, `<xsl:for-each>` و`<xsl:if>`.
تعبيرات XPath الشائعة:

- `book/title`: يحدد عنصر `<title>` داخل عنصر `<book>`.
- `@id`: يحدد الخاصية `id` لعنصر معين.
- `*`: يطابق أي عنصر.

Output Formats

XSLT can transform XML documents into various output formats, including:
HTML: Used for displaying XML data in web browsers.

Example:

```
<xsl:template match="book">
  <h2><xsl:value-of select="title"/></h2>
  <p>Author: <xsl:value-of select="author"/></p>
</xsl:template>
```

Text: Used for generating plain text output.

Example:

```
<xsl:output method="text"/>
```

XML: Used for transforming one XML structure into another.
Example:

```
<xsl:output method="xml" indent="yes"/>
```

تنسيقات الإخراج

يمكن لـ XSLT تحويل مستندات XML إلى تنسيقات متنوعة للإخراج، بما في ذلك:
HTML: يستخدم لعرض بيانات XML في متصفحات الويب.
مثال

Text: تستعمل لتوليد مخرجات بتنسيق نص عادي.
مثال

XML: تستعمل لتوليد مخرجات بتنسيق XML.
مثال

Conclusion

XSLT is a powerful tool for transforming XML documents into various formats, enabling the separation of data and presentation. By mastering XSLT, you can create dynamic and reusable transformations for a wide range of applications, from web development to data integration.

خاتمة

لغة أنماط XSLT أداة قوية لتحويل مستندات XML إلى تنسيقات مختلفة، مما يتيح فصل البيانات عن العرض. بإتقانك XSLT، يمكنك إنشاء تحويلات ديناميكية قابلة لإعادة الاستخدام في تطبيقات متنوعة، من تطوير الويب إلى معالجة البيانات.

XPath

لغة المسارات

XPath (XML Path Language) is a query language used to navigate and select elements, attributes, and text in XML documents. It provides a way to locate nodes in an XML tree using path expressions.

Key Features of XPath:

- **Tree Navigation:** XPath allows traversal of XML nodes in a hierarchical tree structure.
- **Selection Mechanism:** Supports selecting elements, attributes, and text.
- **Filtering & Conditions:** Provides predicates (`[]`) for filtering specific nodes.
- **Operators & Functions:** Supports arithmetic, logical, string, and node functions.

Applications of XPath

- **Web Scraping:** Extracting structured data from HTML/XML.
- **XML Data Processing:** Querying and manipulating XML documents.
- **Automated Testing (Selenium):** Locating web elements in HTML.
- **Configuration File Parsing:** Reading XML-based config files.

XPath (XML Path Language) لغة استعلام تُستخدم للتنقل وتحديد العناصر والسمات والنصوص في مستندات XML. توفر طريقة لتحديد العقد في شجرة XML باستخدام تعبيرات المسار.

الميزات الرئيسية:

- **التنقل في الشجرة:** تسمح XPath بالتنقل بين عقد XML في بنية شجرية هرمية.
- **آلية الاختيار (التحديد):** تدعم اختيار العناصر والسمات والنصوص.
- **التصفية والشروط:** توفر الأقواس [] لتصفية العقد المحددة.
- **المعاملات والدوال:** تدعم العمليات الحسابية والمنطقية والنصية، ووظائف العقد.

تطبيقات XPath

- **استخلاص البيانات من الويب:** استخراج البيانات المنظمة من HTML/XML.
- **معالجة بيانات XML:** استعلام وتعديل مستندات XML.
- **الاختبار التلقائي (Selenium):** تحديد عناصر الويب في HTML.
- **تحليل ملفات الإعدادات:** قراءة ملفات الإعدادات المبنية على XML.

Basic Syntax and Expressions

الصياغة والتعبيرات

Node Selection XPath expressions define paths to select XML nodes.

تحديد العقد تعريف تعبيرات XPath لمسارات تحديد عقد XML.

Expression	Description	الوصف
/	Selects the root node	يحدد العقدة الجذرية
//	Selects nodes anywhere in the document	يحدد العقد في أي مكان في المستند
.	Selects the current node	يحدد العقدة الحالية
..	Selects the parent node	يحدد العقدة الأصلية
@	Selects attributes	يحدد السمات

Path Expressions

Axes in XPath: Axes define relationships between nodes

عبارات المسار

المحاور في XPath:
المحاور تحدد العلاقات بين العقد.

Expression	Description	الوصف
/bookstore/book	Selects all <book> elements inside <bookstore>	يحدد جميع عناصر <book> داخل <bookstore>
//book	Selects all <book> elements	يحدد جميع عناصر <book>
//book	Selects all <book> elements in the document	يحدد جميع عناصر <book>
/bookstore/book[1]	Selects the first <book> element	يحدد العنصر الأول <book>
//book[last()]	Selects the last <book> element	يحدد العنصر الأخير <book>
//book[@category='fiction']	Selects books with category="fiction"	يحدد الكتب ذات الفئة الخيالية

Axis	Description	الوصف
ancestor::	Selects all ancestors of the current node	يحدد جميع الأسلاف للعقدة الحالية
ancestor-or-self::	Selects all ancestors and the current node	يحدد جميع الأسلاف والعقدة الحالية
child::	Selects direct children	يحدد الأبناء المباشرين
descendant::	Selects all descendants	يحدد جميع الفروع
descendant-or-self::	Selects all descendants and the current node	يحدد جميع الفروع والعقدة الحالية
following::	Selects nodes after the current node	يحدد العقد التالية للعقدة الحالية
preceding::	Selects nodes before the current node	يحدد العقد السابقة للعقدة الحالية
self::	Selects the current node	يحدد العقدة الحالية

Example: **مثال**

`//book/ancestor::bookstore` → Selects the <bookstore> ancestor of a <book> element.

يختار العنصر <bookstore> سلف العنصر <book>.

Practical Example مثال عملي

Sample XML Document **مثال مستند**

```xml
<bookstore>
    <book category="fiction">
        <title lang="ar">Hay Ben Yaqdhan</title>
        <author>M. Ibn Tufayl</author>
        <price>29.99</price>
    </book>
    <book category="non-fiction">
        <title lang="ar">Ancient History of Algeria</title>
        <author>M. Elmili</author>
        <price>19.99</price>
    </book>
</bookstore>
```

XPath Queries **استعلامات XPath**

Query استعلام	Description الوصف	Resultنتيجة
`//book/title`	Selects all book titles يحدد جميع عناوين الكتب	`Hay Ben Yaqdhan, Ancient History …`
`//book[@category='fiction']/title`	Selects the title of fiction books يحدد عنوان الكتب الخيالية	`Hay Ben Yaqdhan`
`//book[price>20]/title`	Selects books with price > 20 يحدد الكتب التي سعرها أكبر من 20	`Hay Ben Yaqdhan`

Conclusion

XPath is a powerful tool for navigating XML documents, providing efficient ways to select, filter, and manipulate data. Its integration into programming languages and web technologies makes it a crucial tool for developers working with structured data.

خاتمة

XPath هي أداة قوية للتنقل داخل مستندات XML، حيث توفر طرقًا فعالة لتحديد وتصفية ومعالجة البيانات. تكاملها مع لغات البرمجة وتقنيات الويب يجعلها أداة أساسية للمطورين الذين يعملون مع البيانات المهيكلة.

XQuery

لغة الاستعلامات

XQuery is a powerful query language designed for querying and manipulating XML data. It is often referred to as the "SQL for XML" because it allows users to extract and transform data from XML documents in a way similar to how SQL is used for relational databases. XQuery is a W3C standard and is widely used for querying XML databases, transforming XML data, and integrating XML with other data formats.

Key Features of XQuery:

- **Declarative Language:** XQuery allows users to specify what data they want without describing how to retrieve it.
- **FLWOR Expressions:** Similar to SQL's SELECT-FROM-WHERE, XQuery uses FLWOR (For, Let, Where, Order by, Return) expressions for querying.
- **Supports XPath:** XQuery is built on XPath, enabling powerful navigation and selection of XML nodes.
- **Extensible:** Can be used with XML databases, web services, and other XML-based technologies.

XQuery هي لغة استعلام قوية مصممة للاستعلام ومعالجة بيانات XML. غالبًا ما يُشار إليها باسم "لغة استعلامات لـ XML" لأنها تتيح للمستخدمين استخراج البيانات وتحويلها من مستندات XML بطريقة مشابهة لكيفية استخدام SQL في قواعد البيانات العلائقية. تعد XQuery معيارًا من معايير المؤتمر العالمي مواصفات الوب W3C وتستخدم على نطاق واسع لاستعلام قواعد بيانات XML، وتحويل بيانات XML، ودمج XML مع تنسيقات بيانات أخرى.

الميزات الرئيسية:

- **لغة تصريحية:** تتيح XQuery للمستخدمين اختيار (تحديد) البيانات المطلوبة دون وصف طريقة استرجاعها تقنيا.
- **تعابير FLWOR:** على غرار SELECT-FROM-WHERE في SQL، تستخدم XQuery تعابير FLWOR (For, Let, Where, Order by, Return) للاستعلام.
- **دعم XPath:** تعتمد XQuery على XPath، مما يتيح التنقل القوي وتحديد العقد في XML.
- **قابلة للتوسعة:** يمكن استخدامها مع قواعد بيانات XML وخدمات الويب وغيرها من التقنيات المستندة إلى XML.

Advantages and Disadvantages of XQuery Advantages:

- **Powerful Querying:** Supports complex queries and transformations.
- **Standardized:** A W3C standard, ensuring compatibility across systems.
- **Integration:** Works seamlessly with XML databases and web services.

Disadvantages:

- **Steep Learning Curve:** Requires knowledge of XML, XPath, and FLWOR expressions.
- **Performance:** Can be slower for very large datasets compared to relational databases.

مزايا وعيوب XQuery

مزايا:

- **استعلام قوي:** يدعم استعلامات وتحويلات معقدة.
- **معيارية:** معيار W3C، مما يضمن التوافق مع الأنظمة المختلفة.
- **التكامل:** يعمل بسلاسة مع قواعد بيانات XML وخدمات الويب.

عيوب:

- **منحنى تعلم صعب:** يتطلب معرفة بـ XML وXPath وتعابير FLWOR.
- **الأداء:** أبطأ مع بيانات كبيرة جدًا مقارنة بقواعد البيانات العلائقية.

Applications of XQuery

- **Data Integration:** Combining data from multiple XML sources.
- **Web Services:** Querying and transforming XML data in SOAP-based web services.
- **Content Management:** Extracting and managing content in XML-based CMS systems.
- **Publishing:** Generating reports, books, and other documents from XML data.

Comparison with SQL:

- **XQuery:** Designed for XML data, uses FLWOR expressions, and returns XML or text.
- **SQL:** Designed for relational data, uses SELECT-FROM-WHERE, and returns tables.

When to Use XQuery:

- When working with XML data or XML databases.
- When transforming XML data into other formats (e.g., HTML, JSON).

تطبيقات XQuery

- **دمج البيانات:** دمج البيانات من مصادر XML متعددة.
- **خدمات الويب:** استعلام وتحويل بيانات XML في خدمات الويب المعتمدة على SOAP.
- **إدارة المحتوى:** استخراج المحتوى وإدارته في أنظمة إدارة المحتوى المعتمدة على XML.
- **النشر:** إنشاء التقارير والكتب والمستندات الأخرى من بيانات XML.

المقارنة مع SQL:

- **XQuery:** مصممة لبيانات XML، تستخدم تعابير FLWOR، وترجع بيانات XML أو نصوص.
- **SQL:** مصممة للبيانات العلائقية، تُستخدم SELECT-FROM-WHERE، وترجع جداول.

متى تُستخدم XQuery:

- للعمل مع بيانات XML أو قواعد بيانات XML.
- لتحويل بيانات XML إلى تنسيقات أخرى (مثل HTML، JSON).

Tools for Working with XQuery

أدوات العمل مع XQuery

- **BaseX:** A lightweight XML database with XQuery support.
- **eXist-db:** A native XML database with advanced XQuery capabilities.
- **Saxon:** A Java-based XSLT and XQuery processor.
- **Oxygen XML Editor:** A development tool for writing and testing XQuery.

- **BaseX:** قاعدة بيانات XML خفيفة الوزن تدعم XQuery.
- **eXist-db:** قاعدة بيانات XML أصلية تدعم ميزات XQuery المتقدمة.
- **Saxon:** معالج XSLT وXQuery مبني على جافا.
- **Oxygen XML Editor:** أداة تطوير لكتابة واختبار XQuery.

XQuery Syntax and Structure

XQuery queries are written as expressions that return results in XML, text, or other formats. The basic structure of an XQuery query includes:

- **FLWOR Expressions:** The core of XQuery, used to iterate, filter, sort, and return data.
- **Functions:** Built-in and user-defined functions for performing operations.
- **Variables:** Used to store intermediate results.
- **Conditional Logic:** Supports `if-then-else` statements for conditional processing.

بنية XQuery وصياغتها

تُكتب استعلامات XQuery في تعابير تُرجع نتائج بصيغ XML أو صيغة نصية أو تنسيقات أخرى. تتضمن البنية الأساسية لاستعلام XQuery ما يلي:

- **تعابير FLWOR:** أساس XQuery، تُستخدم للاستعراض والتصفية والترتيب والإرجاع.
- **الدوال:** دوال مدمجة وأخرى يكتبها المستخدم لإجراء العمليات.
- **المتغيرات:** تُستخدم لتخزين النتائج المؤقتة.
- **المنطق الشرطي:** تدعم تعابير "إذا وإلا" `if-then-else` للمعالجة الشرطية.

FLWOR Expressions

FLWOR (For, Let, Where, Order by, Return) expressions are the backbone of XQuery. They allow you to:

- **For:** Iterate over a sequence of nodes.
- **Let:** Bind variables to values.
- **Where:** Filter nodes based on conditions.
- **Order by:** Sort the results.
- **Return:** Construct the output.

Example:

```
for $book in doc("books.xml")/books/book
where $book/price > 30
order by $book/title
return $book/title
```

This query retrieves the titles of books priced above 30, sorted by title.

تعابير FLWOR

تعابير FLWOR (For, Let, Where, Order by, Return) هي أساس لغة لـ XQuery. وتتيح لك:

- **For:** استعراض سلاسل العقد.
- **Let:** إسناد القيم إلى المتغيرات.
- **Where:** تصفية العقد حسب شروط معينة.
- **Order by:** ترتيب النتائج.
- **Return:** بناء المخرجات.

يسترجع هذا الاستعلام عناوين الكتب التي يزيد سعرها عن 30 مرتبة حسب العنوان.

XQuery and XPath

XQuery is built on **XPath**, which is used to navigate and select nodes in an XML document. XPath expressions are embedded within XQuery to specify the data to be queried.

Example:

```
doc("books.xml")/books/book[author="Taha Zerrouki"]
```

This XPath expression selects all books written by "Taha Zerrouki".

الاستعلامات والمسارات

تعتمد XQuery على مسارات XPath، التي تُستخدم للتنقل واختيار العقد في مستند XML. تُضمّن تعابير XPath داخل استعلامات XQuery لتحديد البيانات المطلوبة.

هذا التعبير في XPath يرجع جميع الكتب التي كتبها "Taha Zerrouki".

Functions in XQuery

الدوال

XQuery provides a rich set of built-in functions for string manipulation, mathematical operations, and node handling. Users can also define custom functions.

توفر XQuery قائمة من الدوال المدمجة لمعالجة النصوص، والعمليات الحسابية، والتعامل مع العقد. ويمكن للمستخدمين تعريف دوالهم الخاصة.

Built-in Functions:

الدوال المدمجة:

Function Type	Functions	الدوال	أنواع الدوال
String Functions	`concat(), substring(), contains()`		**دوال النصوص**
Mathematical Functions	`sum(), avg(), round()`		**الدوال الرياضية**
Node Functions	`count(), name(), data()`		**دوال العقد**

Example:

مثال:

```
for $book in doc("books.xml")/books/book
return concat($book/title, " by ", $book/author)
```

This query concatenates the title and author of each book.

هذا الاستعلام يربط عنوان كل كتاب ومؤلفه.

Conditional Logic

المنطق الشرطي

XQuery supports conditional logic using `if-then-else` statements.

يدعم XQuery المنطق الشرطي باستخدام عبارات `if-then-else`.

Example:

مثال:

```
for $book in doc("books.xml")/books/book
return if ($book/price > 50) then <expensive>{$book/title}</expensive>
       else <affordable>{$book/title}</affordable>
```

This query categorizes books as "expensive" or "affordable" based on their price.

يصنّف هذا الاستعلام الكتب حسب سعرها إلى "غالٍ" أو "معقول".

Constructing XML Output

تنسيق ناتج XML

XQuery can construct new XML documents as part of the query result.

يمكن لـ XQuery إنشاء مستندات XML جديدة كجزء من نتيجة الاستعلام.

Example:

```
<books>
{
  for $book in doc("books.xml")/books/book
  return <book>{$book/title, $book/author}</book>
}
</books>
```

This query creates a new XML document with a list of books, including their titles and authors.

يُنشئ الاستعلام مستند XML جديدًا فيه قائمة الكتب، وعناوينها ومؤلفيها.

XQuery and XML Databases

XQuery is commonly used with XML databases like **eXist-db**, **BaseX**, and **MarkLogic**. These databases store XML natively and provide efficient querying capabilities using XQuery.

Example: Querying an XML Database

```
for $employee in collection("employees")/employees/employee
where $employee/salary > 50000
return $employee/name
```

This query retrieves the names of employees with a salary greater than 50,000 from an XML database.

قواعد بيانات XML

يُستخدم XQuery استخدامًا واسعًا مع قواعد بيانات XML مثل **eXist-db** و**BaseX** و**MarkLogic**. تخزن قواعد البيانات هذه XML تخزينًا أصليًّا وتوفر إمكانيات استعلام فعّالة باستخدام XQuery.

مثال استعلام في قاعدة بيانات XML

يسترجع هذا الاستعلام أسماء الموظفين الذين يتقاضون راتبًا أكبر من 50000 من قاعدة بيانات XML.

Conclusion

XQuery is a versatile and powerful language for querying and transforming XML data. Its integration with XPath, support for FLWOR expressions, and ability to construct XML output make it an essential tool for working with XML in databases, web services, and content management systems. By mastering XQuery, you can efficiently extract, manipulate, and integrate XML data for a wide range of applications.

الخاتمة

XQuery لغة قوية متعددة الاستخدامات لاستعلام وتحويل بيانات XML. تكاملها مع XPath، ودعمها لتعابير FLWOR، وقدرتها على إنشاء إخراج XML يجعلها أداة أساسية للعمل مع XML في قواعد البيانات وخدمات الويب وأنظمة إدارة المحتوى. بإتقانك XQuery، يمكنك استخراج بيانات XML ومعالجتها ودمجها بكفاءة في تطبيقاتك.

XML DOM

نموذج كائن المستند

Introduction

مقدمة

The **XML Document Object Model (DOM)** is a programming interface for XML documents. It represents an XML document as a tree structure, where each node corresponds to a part of the document (e.g., elements, attributes, text). The DOM allows programs to dynamically access, manipulate, and update the content, structure, and style of XML documents.

Key Features of XML DOM:

- **Tree Structure:** Represents the XML document as a hierarchy of nodes.
- **Platform- and Language-Independent:** Can be used in various programming languages, including Python, Java, and JavaScript.
- **Dynamic Updates:** Enables real-time manipulation of the XML document.

نموذج كائن مستند XML (DOM) هو واجهة برمجية لمستندات XML. يمثل مستند XML تمثيلا شجريًا، حيث تمثل كل عقدة جزءًا من المستند (مثل العناصر، والسمات، والنصوص). يتيح DOM للبرامج الوصول إلى محتوى مستندات XML وهيكلها وتنسيقها وتحديثها ديناميكيًا.

الميزات الرئيسية لـ XML DOM:

- **بنية شجرية:** يمثل مستند XML على هيئة تسلسل هرمي للعقد.
- **مستقل عن النظام واللغة:** يمكن استخدامه في لغات برمجة مختلفة، مثل Python و Java و JavaScript.
- **التحديث الديناميكي:** يتيح التعامل مع مستند XML في الوقت الفعلي.

Advantages

- Provides a tree-based representation of structured documents, making it intuitive to navigate and manipulate elements.
- Platform- and language-independent standard supported by many tools and libraries.
- Enables full access to the document structure, allowing for comprehensive modifications and queries.

المزايا

- يوفر تمثيلاً شجريًا للمستندات الهيكلية، مما يجعله بديهيًا للتنقل والتعامل مع العناصر.
- معيار مستقل عن المنصات واللغات، مدعوم من قبل العديد من الأدوات والمكتبات.
- يتيح الوصول الكامل إلى بنية المستند، مما يسمح بالتعديلات والاستعلامات الشاملة.

Disadvantages

- Memory-intensive, as the entire document must be loaded into memory, which can be inefficient for large files.
- Can be slower compared to event-driven or stream-based models for certain operations.
- Complex documents may lead to increased processing time and more complicated code for traversing and updating nodes.

العيوب

- يستهلك الكثير من الذاكرة، حيث يجب تحميل المستند كاملة في الذاكرة، مما يجعله غير فعال مع الملفات الكبيرة.
- قد يكون أبطأ مقارنةً بالنماذج المعتمدة على الأحداث أو التدفق في بعض العمليات.
- المستندات المعقدة قد تؤدي إلى زيادة في وقت المعالجة وصعوبة في كتابة الشيفرة الخاصة بالتنقل والتعديل.

DOM Node Types

The DOM tree consists of different types of nodes, each representing a part of the XML document. Common node types include:

- **Document Node:** The root of the DOM tree, representing the entire XML document.
- **Element Node:** Represents an XML element (e.g., `<book>`).
- **Attribute Node:** Represents an attribute of an element (e.g., `id="1"`).
- **Text Node:** Represents the text content of an element (e.g., `Introduction to XML`).
- **Comment Node:** Represents a comment in the XML document (e.g., `<!-- This is a comment -->`).

أنواع العقد

تتكون شجرة DOM من أنواع مختلفة من العقد، حيث تمثل كل عقدة جزءًا من مستند XML. تشمل أنواع العقد الشائعة:

- **عقدة المستند:** تمثل جذر شجرة DOM وتمثل المستند بأكمله.
- **عقدة العنصر:** تمثل عنصر XML (مثل `<book>`).
- **عقدة السمة:** تمثل سمة لعنصر (مثل `"id="1`).
- **عقدة النص:** تمثل محتوى النص داخل عنصر (مثل `Introduction to XML`).
- **عقدة التعليق:** تمثل تعليقًا داخل مستند XML (مثل `<!– This is a comment –>`).

Python's `xml.dom.minidom` Library

The `xml.dom.minidom` module is a lightweight implementation of the DOM interface in Python. It provides a simple way to parse, create, and manipulate XML documents.

Key Features of `minidom`:

- **Easy to Use:** Simple API for basic XML processing tasks.
- **Lightweight:** Suitable for small to medium-sized XML documents.
- **Supports DOM Level 1 and 2:** Provides core DOM functionality.

مكتبة ميني.دوم بلغة بيثون

تعد DOM معيارًا عابرًا للغات البرمجة والمنصات، وقد اخترنا في هذا الكتاب العمل على مكتبة بسيطة بلغة بيثون.
توفر وحدة xml.dom.minidom مكتبة بسيطة لواجهة DOM بلغة بيثون. توفر طريقة سهلة لتحليل وإنشاء والتعامل مع مستندات XML.

الميزات الرئيسية لـ `minidom`:

- **سهولة الاستخدام:** واجهة برمجية بسيطة لمهام معالجة XML الأساسية.
- **بسيطة:** مناسب لمستندات XML الصغيرة إلى المتوسطة الحجم.
- **يدعم DOM المستويين 1 و 2:** يوفر وظائف DOM الأساسية.

Parsing XML with `minidom`

To work with XML using `minidom`, you first need to parse the XML document. This creates a DOM tree that you can navigate and manipulate.

Loading an XML Document Use the `parse()` function to load an XML file.

معالجة مستند XML

للتعامل مع XML باستخدام `minidom`، يجب أولاً تحليل المستند لإنشاء شجرة DOM يمكن التنقل فيها والتعامل معها.

تحميل مستند XML استخدم الدالة `parse()` لتحميل ملف XML.

```
from xml.dom import minidom
```

```
# Parse an XML file
dom = minidom.parse('example.xml')
```

Loading an XML String Use the `parseString()` function to load an XML string.

تحميل سلسلة XML استخدم الدالة `parseString()` لتحميل سلسلة XML.

```
xml_string = '<root><child>Content</child></root>'
dom = minidom.parseString(xml_string)
```

Navigating the DOM Tree

التنقل في شجرة DOM

Once the XML document is parsed, you can navigate the DOM tree using various methods and properties.

بمجرد تحليل مستند XML، يمكنك التنقل في شجرة DOM باستخدام طرق وخصائص مختلفة.

Accessing the Root Element Use the `documentElement` property to get the root element.

الوصول إلى العنصر الجذر استخدم الخاصية `documentElement` للحصول على العنصر الجذر.

```
root = dom.documentElement
print(root.tagName)  # Output: root
```

Accessing Child Nodes Use the `childNodes` property to get a list of child nodes.

الوصول إلى العقد الفرعية استخدم الخاصية `childNodes` للحصول على قائمة بالعقد الفرعية.

```
for child in root.childNodes:
    if child.nodeType == child.ELEMENT_NODE:
        print(child.tagName)
```

Accessing Element Attributes Use the `getAttribute()` method to retrieve attribute values.

الوصول إلى سمات العناصر استخدم الطريقة `getAttribute()` لاسترجاع قيم السمات.

```
element = root.getElementsByTagName('child')[0]
print(element.getAttribute('id'))
```

Manipulating the DOM Tree

التعامل مع شجرة DOM

`minidom` allows you to modify the DOM tree by adding, updating, or deleting nodes.

يسمح `minidom` بتعديل شجرة DOM عن طريق إضافة أو تحديث أو حذف العقد.

Creating New Elements Use the `createElement()` method to create a new element.

إنشاء عناصر جديدة استخدم الطريقة `createElement()` لإنشاء عنصر جديد.

```
new_element = dom.createElement('newChild')
new_element.appendChild(dom.createTextNode('New Content'))
root.appendChild(new_element)
```

Updating Element Content Use the `firstChild` property and `nodeValue` to update text content.

تحديث محتوى العناصر تُستخدم الخاصية `firstChild` والطريقة `nodeValue` لتحديث المحتوى النصي.

```
element = root.getElementsByTagName('child')[0]
element.firstChild.nodeValue = 'Updated Content'
```

Removing Elements Use the `removeChild()` method to delete an element.

إزالة العناصر استخدم الطريقة `removeChild()` لحذف عنصر.

```
element = root.getElementsByTagName('child')[0]
root.removeChild(element)
```

Writing XML with `minidom`

توليد XML

After manipulating the DOM tree, you can write the updated XML to a file or string.

بعد معالجة شجرة DOM، يمكنك كتابة XML الناتج في ملف أو سلسلة نصية.

Writing to a File Use the `writexml()` method to save the DOM tree to a file.

الكتابة في ملف تُستخدم الطريقة `writexml()` لحفظ شجرة DOM في ملف.

```
with open('output.xml', 'w') as f:
    dom.writexml(f)
```

Writing to a String Use the `toxml()` method to convert the DOM tree to an XML string.

الكتابة في سلسلة نصية استخدم الطريقة `toxml()` لتحويل شجرة DOM في سلسلة نصية XML.

```
xml_string = dom.toxml()
print(xml_string)
```

Example: Full Workflow with `minidom`

مثال كامل

Here's a complete example that demonstrates parsing, navigating, manipulating, and writing an XML document using `minidom`:

فيما يلي مثال كامل يوضح تحليل مستند XML واستعراضه ومعالجته وكتابته بواسطة minidom:

```
from xml.dom import minidom

# Parse an XML file
dom = minidom.parse('example.xml')
root = dom.documentElement

# Access and print the root element
print(f"Root element: {root.tagName}")

# Access child nodes
for child in root.childNodes:
    if child.nodeType == child.ELEMENT_NODE:
        print(f"Child element: {child.tagName}")

# Access and update an element
element = root.getElementsByTagName('child')[0]
print(f"Original content: {element.firstChild.nodeValue}")
element.firstChild.nodeValue = 'Updated Content'

# Add a new element
```

```
new_element = dom.createElement('newChild')
new_element.appendChild(dom.createTextNode('New Content'))
root.appendChild(new_element)

# Write the updated XML to a file
with open('output.xml', 'w') as f:
    dom.writexml(f)

# Print the updated XML as a string
print(dom.toxml())
```

Conclusion

The Document Object Model (DOM) is a standard and comprehensive approach for working with structured documents such as XML and HTML. It represents the document as a tree structure, allowing developers to easily navigate, access, and manipulate elements and nodes. While DOM may not be the most efficient solution for handling large documents due to its memory consumption, it remains a practical choice for small to medium-sized data.

As an example of using DOM in programming, the xml.dom.minidom library in Python offers a simplified interface for XML processing. Although it lacks the performance and advanced features of third-party libraries like lxml, it is well-suited for basic tasks such as parsing, traversing, modifying, and writing XML data.

خاتمة

يُعد نموذج الكائنات المستنداتية (DOM) طريقة قياسية شاملة للتعامل مع المستندات الهيكلية مثل XML وHTML. يسمح هذا النموذج بتمثيل المستند على شكل شجرة، مما يتيح للمطورين إمكانية التنقل والتعديل على العناصر والعُقد بسهولة. ورغم أن استخدام DOM قد لا يكون الأنسب للمستندات الكبيرة بسبب استهلاكه العالي للذاكرة، فإنه يظل خيارًا فعالًا للمستندات الصغيرة والمتوسطة.

كمثال على تطبيق DOM في البرمجة، توفر مكتبة minidom.xml.dom في لغة بايثون وسيلة مبسطة لمعالجة مستندات XML. وعلى الرغم من أنها ليست بالكفاءة نفسها التي توفرها مكتبات خارجية مثل lxml، إلا أنها مفيدة جدًا في المهام الأساسية كتحليل المستندات، التنقل بين العناصر، وإجراء التعديلات ثم حفظ النتائج.

2. Practical works أعمال تطبيقية

Contents

2.1 How to write a report كيفية إعداد تقرير 38
2.2 Practical works No. 1[**Python**] العمل التطبيقي 1 40
2.3 Practical works No. 2[**DTD**] العمل التطبيقي 2 46
2.4 Practical works No. 3[**XSLT**] العمل التطبيقي 3 51
2.5 Practical works No. 4[**XSD**] العمل التطبيقي 4 55
2.6 Practical works No. 5[**XQuery**] العمل التطبيقي 5 61
2.7 Practical works No. 6[**XML DOM**] العمل التطبيقي 6 70

How to write a report كيفية إعداد تقرير

Note : The recommendations for preparing a report must be followed.

Title page

You must specify the following information on the guard page:

- Binomial names
- Specialty and Group
- Module
- Practical work number + Project Title.

Report Structure :

1. **Introduction** : explaining the problem to be solved and the questions asked
2. **Methodology** : the technique used (methods and algorithms)
3. **Tools** : to cite the libraries, frameworks, tools to realize the work.
   Avoid representing tools that are presented in lesson or already in use.
4. **Tests and results**: present examples of input data and results after execution.
5. **Insert code in appendices** : structure the code and provide it with comments explaining the important things.

Remark :

- No more than two people (a pair)
- Use only clamps (without bookbinder).
- Avoid plagiarism (all similar works are rejected and rated to Zero)
- If you want to help your colleague, help him learn.
- Colors are not necessary.

Submission :

- The report must be submitted within the scheduled dates.
- Submission of the report online under the platform.
- The delay in submitting the report is taken into account.
  To submit:
  - The report of the work done during the practical session.
  - Home working.

كيفية إعداد تقرير

ملحوظة : يجب اتباع توصيات إعداد تقرير العمل التطبيقي.

صفحة الغلاف

في **صفحة الغلاف** ، يجب عليك تحديد المعلومات التالية:

- أسماء أعضاء الفريق
- المستوى التخصص / الفوج
- اسم المادة
- رقم العمل التطبيقي و عنوان المشروع.

هيكل التقرير:

1. **مقدمة:** شرح المسألة والمطلوب
2. **منهجية العمل:** التقنية المستخدمة (الطرائق والخوارزميات)
3. **الأدوات:** اذكر المكتبات والأدوات وأطر العمل اللازمة لإنجاز المطلوب.
   تجنب ذكر الأدوات التي سبق تقديمها في الدرس أو سبق استخدامها.
4. **الاختبارات والنتائج:** تقديم أمثلة لبيانات الإدخال والنتائج بعد التنفيذ.
5. **متن البرنامج المصدري في الملحق:** صُغ البرنامج صياغة جيدة، وزوّده بتعليقات تشرح الأشياء المهمة.

ملاحظات

- لا يزيد الفريق عن عضوين اثنين
- أمسك الورق بالدبّاسة (لا داعي لاستعمال تغليف).
- تجنب الغش والانتحال (تُرفض جميع الأعمال المتماثلة وتُصفّر)
- إذا كنت ترغب في مساعدة زميلك ، ساعده على التعلم.
- الطباعة الملونة ليست ضرورية.

التسليم

- يجب تسليم التقرير في المواعيد المحددة.
- تسليمه رقميّا في منصة التعليم الالكتروني. وليس بالبريد الإلكتروني.
- يُحتسب التأخر في التسليم
- يجب تسليم :

– تقرير الأعمال المنفذة خلال حصة الأعمال التطبيقية.
– الواجب المنزلي.

Practical works No. 1[Python] العمل التطبيقي 1

Chapter No. 1:	Python		الفصل 1
Objectives :	Learn Python language	تعلم لغة برمجة بيثون	الأهداف:
Tools:	Python, PyScripter/PyCharm		أدوات

Tuturial تدريب

Work to do **للعمل**

- Install Python under Windows or Linux تثبيت بيثون على وندوز أو لينكس
- Run a Python script تنفيذ برنامج بلغة بيثون
- Write programs with Python كتابة برامج بلغة بيثون

Run the following code نفّذ ما يلي

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# character encoding declaration, often used UTF-8,
# to support all multilingual.
print("Salam Alykom")
#print(u"السلام عليكم")

#  Variables declaration
# no explicit declaration
# just assign a value to specify the type of a variable
age = 125 # integer
print ("type of age", type(age))
salaire = 3.14
print ("type of salaire", type(salaire))
name = "Ahmed"
print ("type of name", type(name))

#substitution
s2 = "I am %s"%name
s3 = "I am %d old "%age
print(s2, s3)


# listes
liste = [1,2,31,4]
print ("type of ls", type(liste))
print (liste)
# add an element
liste.append(5)
print (liste)

# word list

wordlist = ["I", "am", "learning", "english"]
```

```
# Flows Control
a = 15
if a > 15:
    # pay attention to tabs
    print ('a is greater than 15')
elif a <= 15 and a >= 10:
    print ('a between  10 and 15')
else:
    print ('a is less than 10')



# test if an element belongs to a list
a = 5
if a in liste:
    print ("a exists in list")


# Loops
for word in wordlist:
    print (word)

# Loops and tests
# Count occurrences of a character in a string
chaine = """The sky, oh the sky! It stretches above us, vast and endless.
The sky is a canvas, a canvas of blue, blue like the ocean on a clear
summer day. It's a canvas of possibilities, possibilities as vast as
the sky itself."""

caractere="e"
cpt = 0
for c in chaine:
    if c == caractere:
        cpt += 1
print ("Char %s exists %d times"%(caractere, cpt))
```

Run the following code: be careful the code contains errors.

نفّذ البرنامج الآتي، انتبه فيه بعض الأخطاء.

```
def clean_text(text):
     "
        remove punctuation  form a text
     '
     punct = '+.*/?,;\'#
       for  p  in  punct
              text  =  text.replace(p,")
      return   text

def  tokenize(text)
    "
```

```
        Convert  text into tokens,  return  a list of  tokens(words)
        '''
       return  text.split( )
def tokenize2(text)
       '
       Convert  text  into tokens,  return a list  of tokens(words)
        '''
       spaces = r\s+
    words = re.split(spaces,  text)
        return word


def  word_freq(words):

       Counts  words and return a  dictionary of words
       with  their occurrences

        index  = {}
    for w in  words:
             if w in  index
              index[w]  += 1
           else
                index[w] =  1
       return index
def most_common_word(words_freq_table)
       '
    Counts words  and return a  dictionary of  words
    with their  occurrences
      ''
       frequent =
     frequency  = 0
       for  word in  words_freq_table
            if  words_freq_table[word] >  frequency:
              frequent  =  word
                 frequency =  words_freq_table[word]
       return frequent




def read_file(filename)
     ''  Read  a  text from  file

       text =  '''
       try
             fl =  open(filename)
       except:
               print(Can't open  file,  ",  filename)
             sys.exit()
      #  if  success
        text  = fl.read()
       lines = fl.readlines()
      return text

def  main(args):
```

```
    text = ""Surprise steepest recurred landlord mr  wandered amounted
      of. Continuing  devonshire but  considered its.
      Rose  past oh shew  roof is song neat. Do depend  better praise
      do friend garden  an wonder  to.
      Intention age nay otherwise  but  breakfast.
      Around garden beyond  to  extent by.
      "
     text =  clean_text(text)
       print(text)
     #  tokenize  text
      words  = tokenize(text)
     print(words)
       words_nb = word_freq(words)
      print(words_nb)
    data  = read_file("data.txt)
      print(data)
      text  =  data
     words  =  tokenize(text)
      print(words)
      word_freq_table = word_freq(words)
     print(word_freq_table)
      freqw  =  most_common_word(word_freq_table)
       print(most frequent word is',freqw,word_freq_table[freqw])
      return  0
```

Home work واجب منزلي

Realize the following project using Python (without using libraries).

- Writing a python program to read data from a text file.
- The data is stored in form of fields separated by tabs.
- The result is displayed as a web page.

Project :

Management of student mentoring in a module

Data : Student list : Level, specialty, group, student number in the group, first name, family name.
Mentoring :

- Number of sessions done by the group عدد الحصص.
- The number of lessens taken by the student (a justification is accounting as 0.5).
- Test scores (two tests).
- participation scores.
- Exam note / make up exam note علامة الامتحان والاستدراك.

Fields to calculate :

- Assiduity rating: a credit of 3 points, each unjustified absence subtract one point, one justification is accounted for as 0.5. Minimum rating is 0.
- Exclusion: A zero in presence results in an exclusion.

- TD rating: attendance on 3, participation on 2,
- Average of module معدل المادة.

Work to do :

- Calculate student scores.
- Show list of non-admitted students.
- Show student statistics by results per session.

Additional questions (optional)

- Redo the same work with Pandas library.

أنجز المشروع باستخدام بايثون (دون مكتبات خاصة):

- اكتب برنامج بيثون لقراءة البيانات من ملف نصي
- البيانات مخزنة كحقول مفصولة بعلامات جدولة.
- عرض النتيجة كصفحة ويب.

مشروع : إدارة متابعة الطلاب في مادة معينة
المعطيات :
قائمة الطلاب: المستوى ، التخصص ، الفوج، المادة ، رقم الطالب في المجموعة ، اللقب ، الاسم الأول،

المتابعة :

- عدد الحصص المنجزة في الفوج
- عدد الحصص التي حضرها الطالب (يحسب التبرير 0.5)
- علامة الفحوص (فحصان)
- علامة المشاركة.
- علامة الامتحان / الاستدراك

الحقول المطلوب حسابها:

- علامة الحضور: رصيد قدره 3 نقاط ، ويطرح كل غياب غير مبرر نقطة ، ويحسب التبرير 0.5. الحد الأدنى للدرجات 0.
- الإقصاء: علامة صفر في الحضور تقصي.
- علامة الأعمال الموجهة: الحضور من 3 ، المشاركة من 2 ،
- معدل المادة.

العمل المطلوب:

- حساب نتائج الطلاب
- عرض قائمة الطلاب غير المقبولين.
- عرض إحصائيات الطلاب بناءً على نتائج كل دورة (عادية/استدراكية).

أسئلة إضافية (اختياري) :

- إعادة العمل بمكتبة الباندا.

References مراجع

- (W3C.org, 2023) W3Schools Tutorials http://www.w3schools.com.
- د. علاء طعيمة (2022a). بايثون عن طريق الأمثلة. 90 مشروع بايثون مع الكود المصدري تم حلها وشرحها. دار العربي. URL: https://dlarabic.com
- د. علاء طعيمة (Dec. 2022b). كيف تبرمج في بايثون. دار العربي. URL: https://dlarabic.com (visited on 07/22/2026)
- ليزا تاغليفيري (Mar. 20, 2025). البرمجة بلغة بايثون. تعلم البرمجة وكتابة البرامج وتنقيحها بلغة بايثون. Ed. by جميل بيلوني. Trans. by محمد بغات and عبد اللطيف إيمش. أكاديمية حسوب. URL: https://academy.hsoub.com/files/15-%D8%A7%D9%84%D8%A8%D8%B1%D9%85%D8%AC%D8%A9-%D8%A8%D9%84%D8%BA%D8%A9-%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86/ (visited on 07/22/2026)
- تشارلز راسل سيفرانس (2023). بايثون للجميع: التعامل مع البيانات باستخدام لغة بايثون 3. Trans. by منصة إلكترونيكس غو. منصة إلكترونيكس غو. URL: https://www.py4e.com/translations/AR/ar/book_ar_2023_01_01.pdf (visited on 07/22/2026)

Additional exercises تمارين للتعمق

Exercise

Write a python program using the "re" library to:

- Verify an email address.
- Verify a phone number.
- Check the strength of a password.

اكتب برنامجا بلغة بيثون واستعمل مكتبة re للتحقق من صحة عنوان بريد الكتروني، وصلاحية رقم هاتف وقوة كلمة السر.

Practical works No. 2[DTD] العمل التطبيقي 2

Chapter No. 2:	DTD		الفصل 2
Objectives :	Handle XML/DTD	التحكم في XML و ملفات التعريف	أهداف:
Tools:	xml copy editor[1]		أدوات:
Data:	directory.xml		بيانات:

Tuturial تدريب

Questions: Here is the following file directory.dtd

إليك ملف تعريف أنواع البيانات directory.dtd الآتي:

```
<!ELEMENT directory (person*)>
<!ELEMENT person (identity)>
<!ELEMENT identity (familyname,firstname,address)>
<!ELEMENT address (street,zip,city,phone)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT address (tel)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
```

1. Check th validity of the directory-v0.1.xml file against the DTD:

تحقق صلاحية الملف directory-v0.1.xml بالنسبة إلى ملف التعريف DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE directory SYSTEM "directory.dtd">
<directory>
    <person>
        <identity>
            <familyname>Achour</familyname>
            <firstname>Hamza</firstname>
            <address>
                <street>25 avenue de 05 juillet 1962 </street>
                <zip-code>16000</zip-code>
                <city>Alger</city>
                <phone type="mobile">071002500</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>AOmar</familyname>
            <firstname>Ahmed</firstname>
            <address>
                <street>18 avenue du Ibn Badis</street>
                <zip-code>10000</zip-code>
```

```
                <city>Bouira</city>
                <phone type="fix">026942525</phone>
                <phone type="mobile">050252525</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>Alouache</familyname>
            <firstname>Ayoub</firstname>
            <address>
                <street>18 avenue du Didouche Mourad</street>
                <zip-code>42000</zip-code>
                <city>Tipaza</city>
                <phone type="fix">025942525</phone>
                <phone type="mobile">051252525</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <firstname>Razika</firstname>
            <familyname>Khaladi</familyname>
            <address>
                <street>18 Ahmed Ben Aissa</street>
                <zip-code>10000</zip-code>
                <city>Bouira</city>
                <phone type="fix">026666666</phone>
                <phone type="mobile">052525250</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>Benkadour</familyname>
            <firstname>Ahmed</firstname>
            <address>
                <street>18 avenue du Ibn Badis</street>
                <zip-code>10000</zip-code>
        </address>
        </identity>
    </person>
        <person>
                <identity>
            <familyname>Ribiai</familyname>
            <firstname>Hamada</firstname>
            <address>
                <street>18 avenue du Ibn Badis</street>
                        <city>Bouira</city>
                            </address>
        </identity>
    </person>
</directory>
```

- Use XMLcopy editor, menu XML/validate

- Under Notepad++ go to Plugins > Plugin manager > Show Plugin Manager then find Xml Tools plugin. Tick the box and click Install.

- في برنامج Notepad++ إلى إضافات / إدارة الإضافات/ اعرض مدير الإضافات ثم ابحث عن أدوات Xml. اخترها ثم اضغط للتثبيت.
- على لينكس استعمل

```
xmllint --noout --dtdvalid test.dtd test.xml
```

Update DTD, and correct possible errors on the XML file.

حدّث ملف التعريف DTD وصحّح الأخطاء المحتملة في ملف XML.

Question: Draw a tree that represents the document structure according to the DTD file.

سؤال: ارسم شجرة (Tree) تمثل بنية الوثيقة حسب ملف الDTD.

DTD File improvement: Modify the `directory.dtd` file to support the following requirements, making the necessary adjustments to the `directory.xml` file to ensure compatibility with the new DTD.

To do:

- Add an email element to the person's address.
- Make the phone element optional.
- Allow multiple phone numbers for the person.
- Add a country element after the city element.
- Ensure the XML file implements all the above modifications.

تطوير ملف DTD:

عدّل ملف directory.dtd بحيث يدعم المتطلبات التالية، مع إجراء التعديلات اللازمة على ملف directory.xml ليظل متوافقًا مع الDTD الجديد.

المطلوب

- أضف عنصر email إلى عنوان الشخص (address).
- اجعل عنصر phone اختياريًا.
- اسمح بتسجيل أكثر من رقم هاتف للشخص.
- أضف عنصر country بعد عنصر city.
- تأكد من أن ملف XML يحقق جميع التعديلات السابقة.

DTD File Development: Here you have `league.xml` for footbal league, define a DTD for this file.
إليك ملف `league.xml` ، عرّف ملف DTD لرابطة كرة القدم.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<league>
    <match id="M001">
        <competition>Algerian League</competition>
        <round>15</round>
        <date>2026-02-18</date>
        <stadium>5 Juillet</stadium>

        <home-team>
            <name>MC Alger</name>
        </home-team>

        <away-team>
            <name>ES  Setif</name>
        </away-team>

        <score>
            <home>2</home>
            <away>2</away>
        </score>

        <goals>
            <goal>
                <minute>12</minute>
                <team>home</team>
                <player>...</player>
            </goal>

            <goal>
                <minute>65</minute>
                <team>away</team>
                <player>...</player>
            </goal>
        </goals>

        <extra-time>
            <played>yes</played>
            <score>
                <home>1</home>
                <away>0</away>
            </score>
        </extra-time>

        <penalties>
            <home>4</home>
            <away>3</away>
        </penalties>
    </match>
</league>
```

Question: Add more details to the XML files and improve DTD file.

سؤال: أضف تفاصيل أكثر لملف XML، وحسّن ملف تعريف DTD.

Homework العمل المنزلي

Perform the next project using Python and XML.

- Write a python program to read data from a text file.
- The data is stored in the form of fields separated by tabs.
- The result is displayed as an XML document.
- Define the DTD of the document, then validate the resulting document.

أتجز العمل بلغة بيثون وXML

- اكتب برنامج بيثون لقراءة البيانات من ملف نصي
- البيانات مفصولة بعلامة جدولة.
- الناتج يعرض في شكل مستند XML.
- عرّف هيكلة البيانات بوساطة DTD ثم تحقق من صلاحية المستند الناتج.

Project : مشروع

Management of student mentoring in a module (Refer to first practical work)

مشروع متابعة الطلاب في مادة ما، (راجع العمل التطبيقي الأول)

Addtional questions (optional) أسئلة إضافية (اختيارية)

Define a DTD for Quran-juza-amma.xml جزء عمّ [2].

ليكن الملف Quran-juza-amma.xml جزء عمّ، عرّف ملف DTD.

References مراجع

- Quran Data: `https://github.com/linuxscout/DSSBook-resources/`
- Quran Metadata from Tanzil project: `https://tanzil.net/docs/quran_metadata`
- XML copy editor: `https://xml-copy-editor.sourceforge.io`

[2] `https://github.com/linuxscout/DSSBook-resources/`

Practical works No. 3[XSLT] العمل التطبيقي 3

Chapter No. 2:	XSLT		الفصل 2
Objectives :	Handle XML/CSS/XSL	التحكم في XML	الأهداف:
Tools:			أدوات
Data:	directory.xml		بيانات

Tuturial تدريب

CSS example : directory.css

```
Directory{ background-color: #8A1619 }
person {    display: table; }
familyname {     color: #fff;
    font-family: 'Righteous', cursive;
    font-size: 40px;
    line-height: 60px;
    text-transform: uppercase;
    text-shadow: 2px 2px 0 #000, margin: 10px 0 24px;
}
firstname {font-size: 30px;
    color: #DFBF84;
    text-shadow: 1px 1px 0 #000, margin: 10px 0 24px;
    line-height: 50px;
}
address {display: table;
    color: #f6f6f6;
    font-size: 16px;
    line-height: 24px;
}
zip-code { padding: 15px; font-size: 20px; }
phone { color: #e8d4a7;  }
```

Link directory.css style sheet to directory.xml by using the following tag:
اربط ورقة النمط directory.css بالملف directory.xml باستعمال الوسم الآتي

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

- Can we modify the CSS file to display only the last name, first name, and city?
- Can we modify the CSS file to display the first name before the family name?

- هل يمكننا تعديل ملف CSS لعرض الاسم واللقب والمدينة فقط؟
- هل يمكننا تعديل ملف CSS لعرض الاسم قبل اللقب؟

XSLT Example

Create the directory.xsl file

أنشئ ملف directory.xsl .

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<h1 align="center">Directory</h1>
<h2>People List</h2>
<xsl:for-each select="directory/person">
    <h3><xsl:value-of select="identity/familyname"/>
     <xsl:value-of select="identity/firstname"/>
     </h3>
    <p><xsl:value-of select="identity/address"/></p>
<hr/>
    </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Link directory.xsl style sheet to directory.xml by using the following tag:

اربط ملف النمط directory.xsl بملف البيانات directory.xml باستعمال الوسم:

```
<?xml-stylesheet type="text/xsl" href="directory.xsl"? >
```

- Can we modify the XSL file to display only the family name, first name, and city?
- Can we modify the XSL file to display the first name before the family name?

- هل يمكننا تعديل ملف XSL لعرض الاسم واللقب والمدينة فقط؟
- هل يمكننا تعديل ملف XSL لعرض الاسم قبل اللقب؟

Modify the code to display each person as follows:

عدّل الكود ليظهر كل شخص كما يلي:

Family name: Achour
First Name: Hamza
Street: 25 avenue de 05 juillet 1962
City: Alger
Phone:071002500

Modify the code to display a table that contains the list of people, as follows:

عدّل الكود ليظهر جدولا فيه قائمة الأشخاص كما يلي:

Directory

People List

Family Name	First Name	Address
AOmar	Ahmed	18 avenue du Ibn Badis 10000 Bouira 026942525 050252525
Benkadour	Ahmed	18 avenue du Ibn Badis 10000 Bouira
Ribiai	Hamada	18 avenue du Ibn Badis 10000 Bouira
Khaladi	Razika	18 Ahmed Ben Aissa 10000 Bouira 026666666 052525250

Modify the line عدّل السطر :

```
<xsl:for-each select="directory/person">
```

by بـ :

```
<xsl:for-each select="directory/person[identity/address/city='Bouira']">
```

You can see that we can show that people living in Bouira. يمكننا أن نعرض قائمة من يعيشون في مدينة البويرة.

Homework العمل المنزلي

Perform the next project using XML/CSS/XSLT.

- Use a CSS style sheet for the presentation of the mentoring file.
- Use an XSLT style sheet for the presentation of the mentoring file.
- Use an XSLT style sheet to calculate the calculated fields.
- Use an XSLT style sheet to calculate the requested statistics.

أنجز العمل بلغة XML و CSS وXSLT.

- استعمل ورقة أنماط CSS لعرض ملف المتابعة.
- استعمل ورقة أنماط XSL/T لعرض ملف المتابعة.
- استعمل ورقة أنماط XSL/T لحساب الحقول المحسوبة .
- استعمل ورقة أنماط XSL/T لحساب الإحصائيات المطلوبة .

Project : مشروع

Management of student mentoring in a module (Refer to first practical work)

مشروع متابعة الطلاب في مادة ما، (راجع العمل التطبيقي الأول)

Addtional questions (optional) أسئلة إضافية (اختيارية)

Let have the following file Quran-juza-amma.xml جزء عمّ

- Create a CSS style sheet.
- Create a XSL style sheet.
  - Improve the display of sourates.
  - Show the index of the sourates with their attributes (Makkia, Madania, Number of ayats, number of sourate).
  - Add links to audio files for each sourate.
  - Improve display for mobile devices.

ليكن الملف Quran-juza-amma.xml جزء عمّ

- أنشئ ورقة أنماط CSS.
- أنشئ ورقة أنماط XLS/T.
  - حسنّ عرض السور
  - اصنع فهرسا للسور مع خصائصها (مكية، مدنية، عدد الآيات، رقم السورة)
  - أضف روابط لملفات صوتية لكل سورة
  - حسّن العرض للأجهزة المحمولة

References مراجع

s

- Quran Data: `https://github.com/linuxscout/DSSBook-resources/`
- Quran Metadata from Tanzil project: `https://tanzil.net/docs/quran_metadata`
- XSLT Elements Reference: `https://www.w3schools.com/xml/xsl_elementref.asp`

Practical works No. 4[XSD] العمل التطبيقي 4

Chapter No. 2:	XSD		الفصل 2
Objectives :	Handle XML/DTD/CSS/XSL	التحكم في XML Schema	الأهداف:
Tools:	xml copy editor[3]		أدوات
Data:	directory.xml, bank.xml	books.xml, quran-juz-amma.xml	بيانات

Tuturial تدريب

Questions : Here is directory.xsd إليك ملف التعريف directory.xsd

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="directory">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="person" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="identity">
      <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="familyname" type="xsd:string"/>
        <xsd:element name="firstname" type="xsd:string"/>
        <xsd:element name="address">
        <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="street" type="xsd:string"/>
          <xsd:element name="zip-code" type="xsd:string"/>
          <xsd:element name="city" type="xsd:string"/>
           <xsd:element name="phone" type="xsd:string"
           minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      </xsd:complexType>

      </xsd:element>
    </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

1. Check th validity of the directory-v0.1.xml file against the XSD:

   تحقق صلاحية الملف directory-v0.1.xml بالنسبة إلى ملف التعريف XSD:

   - Use XMLcopy editor, menu XML/validate

- Under Notepad++ go to Plugins > Plugin manager > Show Plugin Manager then find Xml Tools plugin. Tick the box and click Install.
- في برنامج Notepad++ إلى إضافات / إدارة الإضافات/ اعرض مدير الإضافات ثم ابحث عن أدوات Xml. اخترها ثم اضغط للتثبيت.
- Under linux : على لينكس استعمل

```
xmllint --schema yourxsd.xsd yourxml.xml --noout
```

2. Correct possible errors on the XML file. صحّح الأخطاء المحتملة في ملف XML.

3. Update the XSD to add attributes for phone element. حدّث ملف التعريف XSD لإضافة خصائص لعنصر الهاتف.

```
<xsd:element name="phone" minOccurs="0" maxOccurs="unbounded">
<xsd:complexType>
<xsd:simpleContent>
  <xsd:extension base="xsd:string">
  <xsd:attribute name="type" type="xsd:string"/>
  </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

Exercise 2 : Define the DTD and XML schema for the bank.xml file to meet the following requirements:
أنشئ ملف تعريف DTD ومخطط XSchema لملف bank.xml لتلبية المتطلبات التالية:

- There are two types of accounts: checking and savings accounts حساب جاري أو إدخار .
- the account ID is unique.
- the client ID is unique.
- `'c_id'` refers to the client and `'ac_id'` to the account.
- The credit must be greater than -5000.
- Use inheritance to derive current (checking) and savings accounts from a basic account type.

- الحسابات نوعان: حسابات جارية وحسابات إدخار.
- معرف الحساب (ID) وحيد.
- معرف الزبون (ID) وحيد.
- 'c_id' يشير إلى الزبون و'ac_id' يشير إلى الحساب.
- يجب أن يكون الرصيد أعلى من -5000 .
- استخدم الوراثة لاشتقاق الحسابات الجارية (checking) وحسابات الإدخار (saving) من نوع الحساب الأساسي.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<bank>
  <accounts>
    <savings_accounts>
      <savings_account id="a1" interest="0.03">
        <balance>2500</balance>
      </savings_account>
      <savings_account id="a2" interest="0.03">
        <balance>15075</balance>
      </savings_account>
    </savings_accounts>
    <checking_accounts>
      <checking_account id="a3">
        <balance>4025</balance>
      </checking_account>
      <checking_account id="a4">
        <balance>-125</balance>
      </checking_account>
      <checking_account id="a5">
        <balance>325</balance>
      </checking_account>
    </checking_accounts>
  </accounts>
  <customers>
    <customer id="c1">
      <name>Ahmed Ben Rachidon</name>
      <address>Chaabet Yakhlaf, Bouira</address>
    </customer>
    <customer id="c2">
      <name>Marzouk Wail</name>
      <address>M'chadallah, BP 205</address>
    </customer>
    <customer id="c3">
      <name>Anfel Saidi</name>
      <address>Lakhdaria, Krariche 15b</address>
    </customer>
  </customers>
  <customer_accounts>
    <customer_account c_id="c1" ac_id="a2"/>
    <customer_account c_id="c1" ac_id="a3"/>
    <customer_account c_id="c2" ac_id="a4"/>
    <customer_account c_id="c3" ac_id="a1"/>
    <customer_account c_id="c3" ac_id="a5"/>
  </customer_accounts>
</bank>
```

Exercise 3 : Create an XML Schema for the document `Book.xml`

انشئ ملف XML Schema للمستند Book.xml.

Specifications مواصفات

1. the book item must have one or more books (one at least)
2. Each book must contain the following items in sequence: title, edition, author (one or more), publishedDate, type, and price.

3. Each book must contain an ISBN as a mandatory key attribute.
4. ”title” contains any character (type string).
5. ”title” contains a ”type” attribute that takes only the values ”P” and ”H”.
6. ”edition” is an optional element, and must contain a number not exceeding one byte.
7. There may be one or more authors (at least one author) for each book. Each author must have the following child elements: First, middle, Last. Items must be in the order First, middle, and Last, but the Middle is an optional element. The First, Middle, and Last elements are all string types, but the Middle must have no more than one character.
8. The publishedDate month element is defined as a type of unsignedByte with a minimum inclusive value of 1 and a maximum value of 12.
9. The publishedDate day element is defined as a type of unsignedByte with a minimum inclusive value of 1 and a maximum value of 31.
10. The ”publishedDate year” element is defined as a type of gYear.
11. The ”type” element is a string of characters and must be either ”fiction” or ”nonfiction” to be valid.
12. The price element is defined as a decimal type with only three characters to the left of the decimal comma and two to the right. The amount must be greater than zero.

1. يجب أن يحوي عنصر الكتب على كتاب واحد أو أكثر (واحد على الأقل).
2. يجب أن يحتوي كل كتاب على العناصر التالية بالتسلسل: العنوان، الطبعة، المؤلف (واحد أو أكثر)، تاريخ النشر، النوع، والسعر.
3. يجب أن يحتوي كل كتاب على رقم ISBN (الرقم الدولي للكتاب) كخاصية رئيسية إجبارية.
4. ”العنوان” سلسلة حروف.
5. ”العنوان” يحتوي على خاصية ”type” تأخذ قيمتين ”P” و”H” فقط.
6. ”edition” الطبعة عنصر اختياري، وهو عدد لا يتجاوز بايت واحد.
7. للكتاب مؤلف واحد على الأقل. في اسم المؤلف نضع ثلاث عناصر، الاسم الأول، الأوسط والأخير بالترتيب. كل أقسام الاسم سلاسل حروف. الاسم الأوسط اختياري وبه حرف على الأكثر.
8. في ”تاريخ النشر” نعرّف ثلاث قيم (الشهر، اليوم، السنة) العنصر شهر Month نوعه unsignedByte بقيمة دنيا 1 وقيمة قصوى 12.
9. اليوم من نوع unsignedByte ما بين 1 و 31.
10. السنة نعرّفها بالنوع gYear (السنة الغريغورية).
11. العنصر ”النوع” عبارة عن سلسلة حروف تأخذ قيمتين فقط هما خيالي ”fiction” أو واقعي ”nonfiction”.
12. نعرّف السعر بعدد عشري ذو ثلاثة أرقام قبل الفاصلة، ورقمين بعدها، ولا يكون معدومًا.

```xml
<?xml version="1.0" standalone="no"?>
<books>
<book isbn="AB9781418860646">
   <title type="P">XML Comprehensive Step by Step</title>
   <edition>2</edition>
   <author>
      <first>Ahmed</first>
      <middle></middle>
      <last>Amara</last>
   </author>
   <publishedDate>
      <month>7</month>
      <day>9</day>
      <year>2007</year>
   </publishedDate>
   <type>nonfiction</type>
   <price>70</price>
</book>
<book isbn="AB9780132240857">
   <title type="H">Systems Analysis and Design for Dummies</title>
   <edition>7</edition>
   <author>
      <first>Kemal</first>
      <middle>B</middle>
      <last>Kendali</last>
   </author>
   <author>
      <first>Djamel</first>
      <middle>T</middle>
      <last>Kendali</last>
   </author>
   <publishedDate>
      <month>3</month>
      <day>27</day>
      <year>2007</year>
   </publishedDate>
   <type>nonfiction</type>
   <price>166.67</price>
</book>
<book isbn="AB9780805001112">
   <title type="P">The World Map Trap</title>
   <author>
      <first>Peri</first>
      <middle>F</middle>
      <last>Rais</last>
   </author>
   <publishedDate>
      <month>9</month>
      <day>15</day>
      <year>1986</year>
   </publishedDate>
   <type>fiction</type>
   <price>14</price>
</book>
```

```
</books>
```

Homework العمل المنزلي

Use XML schema to define Student mentoring file. استعمل مخطط Xschema لتعريف بيانات تسيير الطلبة ومتابعتهم.

Projet : مشروع

Management of student mentoring in a module (Refer to first practical work)
مشروع متابعة الطلاب في مادة ما، (راجع العمل التطبيقي الأول)

Addtional questions (optional) أسئلة إضافية (اختيارية)

Use XML schema to define Quran-juza-amma.xml جزء عمّ file.
عرّف هيكلة ملف القرآن الكريم بصيغة XschemaXschema

References مراجع

- Quran Metadata from Tanzil project: `https://tanzil.net/docs/quran_metadata`
- XML copy editor: `https://xml-copy-editor.sourceforge.io`
- XML Schema Elements Reference: `https://www.w3schools.com/xml/schema_intro.asp`
- xsd-cheat-sheet `https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/xsd-cheat-sheet.md`
- xsd-cheat-sheet `https://github.com/davidabram/xml-schema-cheat-sheet/blob/master/cheat-sheet.pdf`
- XML Schema - Structures Quick Reference `https://www.cheat-sheets.org/saved-copy/SchemaStructuresQR-2.pdf`

Practical works No. 5[XQuery] العمل التطبيقي 5

Chapter #2:	XQuery		الفصل 2
Objectives :	Handle XML Query	معالجة استعلامات XML	الأهداف :
Tools:	BaseX, BaseXGUI	python saxonche library	أدوات
Data:	directory.xml		بيانات

Tuturial تدريب

Introduction and Relation to SQL **XQuery** is used to query **XML** data in much the same way that **SQL** is used to query relational databases

تُستخدم لغة XQuery للاستعلام عن بيانات XML بنفس طريقة استخدام SQL مع قواعد البيانات العلائقية.

SQL Concept	XQuery (FLWOR)	Purpose	الهدف
`FROM`	**For**	Iterate over XML nodes	التكرار على عناصر XML
`DECLARATION`	**Let**	Define variables and intermediate values	تعريف المتغيرات والقيم الوسيطة
`WHERE`	**Where**	Filter the selected nodes	تصفية النتائج وفق شروط
`ORDER BY`	**Order by**	Sort the results	ترتيب النتائج تصاعديًا أو تنازليًا
`SELECT`	**Return**	Construct the output document	بناء وإرجاع النتيجة

Development Environment and Tools **الأدوات وبيئات التطوير**

Tool	Type	Recommended Use
BaseX GUI	Native XML database engine محرك قاعدة بيانات XML	Query execution, and performance evaluation التطبيق وتنفيذ الاستعلامات
VS Code	Source code editor	Writing and maintaining XQuery programs
+ XML/XQuery Extensions	محرر شيفرة	كتابة وتطوير البرامج
Online XQuery Testers	Web-based testing tool أداة اختبار عبر الويب	Quick experiments during lectures التجارب السريعة أثناء الدروس
Saxon-CHE (Python)	Programming library مكتبة برمجية	Executing XQuery from Python applications دمج استعلامات XQuery داخل برامج Python

Sample XML Document: We will adopt the following people directory file containing advanced real-world situations (changed order of items, missing fields, multiple phones):

In BaseXGUI, create a new database, and place the 'directory.xml' file inside the database folder.

سنعتمد ملف دليل الأشخاص التالي الذي يحتوي على حالات واقعية متقدمة (ترتيب متغير للعناصر، حقول مفقودة، هواتف متعددة):

في برنامج BaseXGUI، أنشئ قاعدة بيانات جديدة، ثم ضع الملف 'directory.xml' داخل مجلد قاعدة البيانات.

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

```xml
<person id="usr-101">
    <identity>
        <familyname>Achour</familyname>
        <firstname>Hamza</firstname>
        <address>
            <street>25 avenue de 05 juillet 1962</street>
            <zip-code>16000</zip-code>
            <city>Alger</city>
            <phone type="mobile">071002500</phone>
        </address>
    </identity>
</person>

<person id="usr-102">
    <identity>
        <familyname>AOmar</familyname>
        <firstname>Ahmed</firstname>
        <address>
            <street>18 avenue du Ibn Badis</street>
            <zip-code>10000</zip-code>
            <city>Bouira</city>
            <phone type="fix">026942525</phone>
            <phone type="mobile">050252525</phone>
        </address>
    </identity>
</person>

<person id="usr-103">
    <identity>
        <familyname>Alouache</familyname>
        <firstname>Ayoub</firstname>
        <address>
            <street>18 avenue du Didouche Mourad</street>
            <zip-code>42000</zip-code>
            <city>Tipaza</city>
            <phone type="fix">025942525</phone>
            <phone type="mobile">051252525</phone>
        </address>
    </identity>
</person>

<!-- اختلاف ترتيب العناصر الداخلية -->
<person id="usr-104">
    <identity>
        <firstname>Razika</firstname>
        <familyname>Khaladi</familyname>
        <address>
            <street>18 Ahmed Ben Aissa</street>
            <zip-code>10000</zip-code>
            <city>Bouira</city>
            <phone type="fix">026666666</phone>
            <phone type="mobile">052525250</phone>
        </address>
    </identity>
</person>
```

```
<!-- غياب حقل المدينة وحقل الهاتف -->
<person id="usr-105">
    <identity>
        <familyname>Benkadour</familyname>
        <firstname>Ahmed</firstname>
        <address>
            <street>18 Cité Thawra</street>
            <zip-code>10000</zip-code>
        </address>
    </identity>
</person>

<!-- غياب الرمز البريدي والهواتف -->
<person id="usr-106">
    <identity>
        <familyname>Ribiai</familyname>
        <firstname>Hamada</firstname>
        <address>
            <street>210 cité Zerrouki</street>
            <city>Bouira</city>
        </address>
    </identity>
</person>

<!-- تعدد الهواتف المحمولة -->
<person id="usr-107">
    <identity>
        <familyname>Brahimi</familyname>
        <firstname>Yacine</firstname>
        <address>
            <street>05 Boulevard Wilaya</street>
            <zip-code>31000</zip-code>
            <city>Oran</city>
            <phone type="mobile">0661112233</phone>
            <phone type="mobile">0770998877</phone>
            <phone type="work">041223344</phone>
        </address>
    </identity>
</person>

<!-- غياب الهواتف كلياً مع وجود بريد إلكتروني -->
<person id="usr-108">
    <identity>
        <familyname>Zidane</familyname>
        <firstname>Karim</firstname>
        <address>
            <street>12 Rue Gaza Ali Mendjeli</street>
            <zip-code>25000</zip-code>
            <city>Constantine</city>
            <email>k.zidane@example.dz</email>
        </address>
    </identity>
</person>
```

```
    <person id="usr-109">
        <identity>
            <familyname>Achour</familyname>
            <firstname>Sonia</firstname>
            <address>
                <street>25 avenue de 07 octobre 2023</street>
                <zip-code>16000</zip-code>
                <city>Alger</city>
                <phone type="mobile">0799887766</phone>
            </address>
        </identity>
    </person>
</directory>
```

FLWOR Basics and Expressions : الأساسيات وتعابير FLWOR

Example 1: Filtering and Clean Merging

Task: Extract the full name and mobile devices of residents in the city of `Bouira`, sorted alphabetically by last name.

مثال 1: الفلترة والدمج النظيف

المطلوب: استخراج الاسم الكامل والأجهزة المحمولة للقاطنين في مدينة Bouira، مرتبين أبجدياً حسب الاسم العائلي.

```
for $p in db:open("tpxq", "directory.xml")/directory/person
where $p/identity/address/city = "Bouira"
order by $p/identity/familyname
return
  <contact id="{ $p/@id }">
    <fullName>{ concat($p/identity/firstname, " ",
                       $p/identity/familyname) }</fullName>
    <mobile>{ $p/identity/address/phone[@type='mobile']/text() }</mobile>
  </contact>
```

Handling Special Cases and Missing Fields

Example 2: Handling Missing Fields (`if-then-else`)

Task: Extract all persons. If the `city` field is missing, output `"N/A"` instead.

معالجة الحالات الخاصة والحقول المفقودة

المطلوب: استخراج كافة الأشخاص، وفي حال عدم وجود قيمة لحقل المدينة (city)، يتم طباعة "N/A".

```
for $p in doc("directory.xml")/directory/person
let $cityName := $p/identity/address/city
return
  <resident>
    <name>{ concat($p/identity/firstname, " ",
                   $p/identity/familyname) }</name>
    <city>
      {
        if (exists($cityName)) then
          $cityName/text()
        else
          "N/A"
```

```
      }
    </city>
  </resident>
```

Question: Modify the query so that it returns the available contact method. If a mobile phone number exists, output it; otherwise, if an email address exists, output it; otherwise, output `"No Contact Method"`.

سؤال: عدّل الاستعلام بحيث يعيد وسيلة الاتصال المتاحة: إذا وُجد هاتف محمول يطبعه، وإلا إذا وجد بريد إلكتروني يطبعه، وإلا يطبع `"No Contact Method"`.

Example 3: Handling Multiple Mobile Phone Numbers (`Quantifiers & count`)
Task: Extract all persons who have **more than one** mobile phone number (`mobile`).

مثال 3: التعامل مع تعدد الهواتف المحمولة المطلوب: استخرج قائمة أشخاص لديهم أكثر من هاتف محمول واحد (`mobile`).

```
for $p in doc("directory.xml")/directory/person
let $mobiles := $p/identity/address/phone[@type='mobile']
where count($mobiles) > 1
return
  <multiMobileUser id="{ $p/@id }">
    <name>{ concat($p/identity/firstname, " ",
    $p/identity/familyname) }</name>
    <mobileNumbers>{ string-join($mobiles, " / ") }</mobileNumbers>
  </multiMobileUser>
```

Advanced Joins and Grouping (`Group By & Self-Join`)

Example 4: Grouping by City (`Group By`)
Task: Group persons by their city and display the number of residents in each city.

الربط والتجميع المتقدم :
المطلوب: تجميع الأشخاص حسب مدنهم مع إظهار عدد السكان في كل مدينة.

```
for $p in doc("directory.xml")/directory/person
let $city := if ($p/identity/address/city)
               then $p/identity/address/city
               else "Unknown"
group by $city
return
  <cityGroup name="{ $city }">
    <totalResidents>{ count($p) }</totalResidents>
    <names>
      {
        for $person in $p
        return <person>{ concat($person/identity/firstname, " ",
                                $person/identity/familyname) }</person>
      }
```

```
    </names>
  </cityGroup>
```

Question: Group people by phone type.

جمّع الأشخاص حسب نوع الهاتف.

Example 5: Finding People Living on the Same Street (`Self-Join`).

Task: Join the document with itself to find any two persons who live on the **same street and in the same city**.

المطلوب: ربط الملف بنفسه لإيجاد أي شخصين يقيمان في نفس الشارع والمدينة.

```
for $p1 in doc("directory.xml")/directory/person,
    $p2 in doc("directory.xml")/directory/person
where $p1/@id < $p2/@id
  and $p1/identity/address/street = $p2/identity/address/street
  and $p1/identity/address/city = $p2/identity/address/city
return
  <neighbors street="{ $p1/identity/address/street/text() }">
    <resident1>{ concat($p1/identity/firstname, " ",
                        $p1/identity/familyname) }</resident1>
    <resident2>{ concat($p2/identity/firstname, " ",
                        $p2/identity/familyname) }</resident2>
  </neighbors>
```

Note : If does'nt work, modify data.

Question: Find people with same names.

ملحوظة: إذا لم تعمل، عدّل البيانات، لإعطاء مثال عن ذلك.

سؤال: ابحث عن أشخاص بنفس الاسم.

Python Integration The `saxonche` library is used to execute XQuery queries from within Python applications.

Start by installing the `saxonche` library.

التكامل مع لغة بيثون تُستخدم مكتبة `saxonche` لتشغيل استعلامات XQuery داخل تطبيقات بيثون.

ابدأ بتثبيت مكتبة `saxonche`.

```
pip install saxonche
```

Run the following code:

نفّذ البرنامج الآتي:

```
from saxonche import PySaxonProcessor

with PySaxonProcessor(license=False) as proc:
    xquery_proc = proc.new_xquery_processor()
    xquery_proc.set_context(file_name="directory.xml")

    #Inquiry for residents of Algiers
```

```
query = f"""
for $p in /directory/person
where $p/identity/address/city = 'Alger'
return concat($p/identity/firstname, ' ', $p/identity/familyname)
"""

xquery_proc.set_query_content(query)

result = xquery_proc.run_query_to_string()

print("Algiers residents:")
print(result)
```

Question : Redo previous tasks and queries using `Python` and `Saxonche` library.
أعد تنفيذ الاستعلامات السابقة، بواسطة برنامج بيثون.

Homework العمل المنزلي

Using `XQuery`, write the necessary queries to compute and extract the required results from the previously developed Student Monitoring Project. Your queries should:

- Compute the Assiduity Rating.
- Compute the Tutorial (TD) Grade.
- Compute the Final Course Grade.
- Extract the list of excluded students.
- Extract the list of students who did not pass.
- Compute summary statistics of the results.
- Generate an XML report containing all the previous results.

To Do

1. Execute all queries using `BaseX`.
2. Execute the same queries using the `saxonche` library through a `Python` program.
3. Submit the `XQuery` files, the `Python` program, and the generated `XML` report.

باستعمال لغة `XQuery`، أنجز الاستعلامات اللازمة لحساب واستخراج النتائج المطلوبة في مشروع متابعة الطلبة المنجز سابقًا، وذلك من أجل:

- حساب علامة المواظبة (Assiduity Rating).
- حساب علامة الأعمال الموجهة (TD).
- حساب المعدل النهائي للمادة.
- استخراج قائمة الطلبة المقصيين.
- استخراج قائمة الطلبة غير الناجحين.
- حساب إحصائيات النتائج.

- إنشاء تقرير بصيغة XML يتضمن النتائج السابقة.

المطلوب:

1. تنفيذ الاستعلامات باستخدام **BaseX**.
2. تنفيذ الاستعلامات نفسها باستخدام مكتبة **(SaxonC (`saxonche`** من خلال برنامج بلغة **Python**.
3. إرفاق ملفات **XQuery**، وبرنامج **Python**، وملف **XML** الناتج (التقرير).

Addtional questions (optional) أسئلة إضافية (اختيارية)

Use Xquery for those tasks applied on Quran XML file `Quran-juza-amma.xml`:

1. Write an XQuery query that extracts the names of all surahs.
2. Extract the names of the surahs that contain more than 20 verses.
3. Sort the surahs in descending order according to the number of verses.
4. Find the longest and the shortest surah.
5. Extract all verses containing a given word (for example: `Allah`, `Al-Nas`, or `Al-Jannah`).
6. Count the number of occurrences of a given word in the entire document.
7. Group the surahs according to their number of verses: fewer than 10, from 10 to 20, and more than 20.
8. Create an XML report containing, for each surah, its name, number, number of verses, first verse, and last verse.
9. Modify the previous report to include only the surahs that satisfy a condition of your choice.
10. Execute all the previous queries using both BaseX and the `saxonche` Python library.

استخدم Xquery لتطبيق المهام الآتية على ملف XML الخاص بالقرآن الكريم `Quran-juza-amma.xml`:

1. استخرج أسماء جميع السور.
2. استخرج أسماء السور التي تحتوي على أكثر من 20 آية.
3. رتّب السور تنازليًا حسب عدد الآيات.
4. استخرج أطول سورة وأقصر سورة من حيث عدد الآيات.
5. استخرج جميع الآيات التي تحتوي على كلمة معينة (مثل: الله، الناس، الجنة).
6. احسب عدد مرات ظهور كلمة معينة في الملف.
7. صنّف السور حسب عدد الآيات إلى: أقل من 10، ومن 10 إلى 20، وأكثر من 20 آية.
8. أنشئ تقريرًا بصيغة XML يتضمن لكل سورة: اسمها، رقمها، عدد آياتها، أول آية، وآخر آية.
9. عدّل التقرير السابق بحيث يضم فقط السور التي تحقق شرطًا معينًا.
10. نفّذ جميع الاستعلامات السابقة باستخدام كل من BaseX ومكتبة `saxonche` في لغة Python.

References مراجع

- Quran Metadata from Tanzil project: `https://tanzil.net/docs/quran_metadata`
- BaseX — Thigh-performance XML database engine : `https://basex.org/basex/gui/`
- Official Saxonica python package for the SaxonC-HE 13.0.0 processor: for XSLT 3.0, XQuery 3.1, XPath 3.1 and XML Schema processing. : `https://pypi.org/project/saxonche/`

Practical works No. 6[XML DOM] العمل التطبيقي 6

Chapter No. 2:	XML DOM		الفصل 2
Objectives :	Handle XML with DOM API	معالجة XML برمجيّا	الأهداف:
Tools:	Python DOM API		أدوات
Data:	directory.xml		بيانات

Tuturial تدريب

Questions : Here is a Python script إليك برنامج بيثون الآتي.

```
#!/usr/bin/env python
#- * -coding: utf - 8 - * -##testdom2.py##
DATA_FILE = 'TP/directory-v0.2.xml'
import xml.dom.minidom as minidom
import sys
def main():
  try:
     xmldoc = minidom.parse(DATA_FILE)
  except:
     print "Can't Open the file",
     sys.exit()
  print xmldoc.toxml()
  return 0
if __name__ == '__main__':
  main()
```

What does this script do?
Create the following function, execute it in the main function:

ماذا يفعل هذا البرنامج؟
أنشئ الدالة التالية ونفذّها في الدالة الرئيسية main:

```
1  def treat_doc(xmldoc):
2      # get the directory list
3      directory = xmldoc.getElementsByTagName('directory')[0]
4      print directory
5      cpt = 0
6      # display person by person
7      for person  in directory.childNodes:
8          print "-"*40
9          print "person n°", cpt
10         print person.toxml()
11         cpt += 1
```

Explain the results
We want to display all the phone numbers in our file, we use the function

اشرح النتائج
نريد عرض جميع أرقام الهواتف في ملفنا، استخدم الدالة

```
def display_tel(xmldoc):
    # display only telephones
     # get the tel list
    telephones = xmldoc.getElementsByTagName('phone')
    print telephones
    cpt = 0
    # display tel by tel
    for tel  in telephones:
        print "-"*40
        print "Tel n°", cpt
        print tel.toxml()
        print "N°:",tel.firstChild.data
        print "Type:",tel.getAttribute("type")
        cpt += 1

```

Show phone numbers by person اعرض أرقام الهاتف حسب الشخص

```
def display_tel_person(xmldoc):
    #get the persons list
    persons = xmldoc.getElementsByTagName('person')
    print persons
    cpt = 0
    # display telephone by person
    for person  in persons:
        print "-"*40
        print "person n°", cpt
        familyname    = person.getElementsByTagName('familyname')[0]
        firstname = person.getElementsByTagName('firstname')[0]
        tels    = person.getElementsByTagName('phone')
        print "*"*20
        print "Familyname:\t",familyname.firstChild.data
        print "Firstname:\t",firstname.firstChild.data
        for tel in tels:
            print "-"*20
            print "N°:",tel.firstChild.data
            print "Type:",tel.getAttribute("type")
        cpt += 1
```

Add new attribute to person element أضف خاصية جديدة للعنصر person

```
def add_id_person(xmldoc):
    #get the persons list
    persons = xmldoc.getElementsByTagName('person')
    print persons
    cpt = 0
    # display person by person
    for person  in persons:
        print "-"*40
        print "person n°", cpt, person.nodeValue, person.nodeType,
        person.setAttribute('id', str(cpt))
        cpt += 1
```

```
12          print person.toxml()
```

Part 2

Give the following code DOM tree ارسم شجرة نموذج المستند DOM للكود الآتي

```
<html>
   <body>
      <div>
         <b>Familyname:</b> <span id="FirstName"></span><br/>
         <b>Firstname:</b> <span id="LastName"></span><br/>
         <b>Tel:</b> <span id="phone"></span><br>
         <b>city:</b> <span id="town"></span>
      </div>
</body>
</html>
```

Add the following javascript code, execute it. أضف كود جافا سكريبت التالي، ثم نفّذه

```
 <script>
         //modify the HTML
         document.getElementById("firstname").innerHTML= "Taha";
         document.getElementById("lastname").innerHTML= "Zerrouki";
      </script>
```

Modify the javascript code to display the phone and the city عدّل كود جافاسكريب الآتي لعرض الهاتف والمدينة.

```
//Create a <p> element
       var para = document.createElement("H1");
// Create a text node
        var t = document.createTextNode("This is a paragraph");
 // Append the text to <p>
        para.appendChild(t);
       para.setAttribute("align","center");
        document.body.appendChild(para);
```

The code that allows you to load an XML file الكود الذي يحمّل ملف xml

N.B. To avoid any security issues when opening a file from the system, please run the file through a web server like Wamp or Xamp.

لتجنب أي مشاكل أمنية تتعلق بفتح ملف من النظام، يرجى تشغيل الملف من خلال خادم ويب مثل Wamp أو Xamp .

```
<script>
    if (window.XMLHttpRequest)
    //if browser supports XMLHttpRequest
    {// Create an instance of XMLHttpRequest object.
    // code for IE7+, Firefox, Chrome, Opera, Safari
     xmlhttp = new XMLHttpRequest();
    }
```

```
    else
    {// code for IE6, IE5
     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    // sets and sends the request for calling "annuaire.xml"
    try {
      xmlhttp.open("GET","directory-v0.2.xml",async = false);
      xmlhttp.send();
       }
    catch(ex){
       alert("Exception: Can't open xml file");
       alert(ex.message);
      }

    // sets and returns the content as XML DOM
    xmlDoc=xmlhttp.responseXML;
    // parseing dom
</script>
```

Retrieve the data استخرج البيانات

```
//parsing the DOM object
  document.getElementById("firstname").innerHTML  =
    xmlDoc.getElementsByTagName("familyname")[0].childNodes[0].nodeValue;
</script>
```

Display first name, phone and city. اعرض الاسم والهاتف والمدينة

Display information in a table اعرض المعلومات في جدول

```
<table  id="table_personnes" border=1>
<tr> <th>Familyname</th><th>Firstname</th><th>Phone</th><th>City</th></tr>
</table>
```

Displaying information in a table عرض المعلومات في جدول

```
var identities = xmlDoc.getElementsByTagName("identity")
var table = document.getElementById("table_person")
 for(i in identities)
  {
  var tr = document.createElement("tr");          // create  a row
  var td = document.createElement("td");          // create  a row
  var familynames  = identities[i].getElementsByTagName("familyname");
  td.innerHTML = familynames[0].childNodes[0].nodeValue;
    tr.appendChild(td);       // add td to tr
     // redo for another data
     tr.appendChild(td);
     table.appendChild(tr);
}
```

Display first name, phone and city. اعرض الاسم والهاتف والمدينة

Homework العمل المنزلي

Redo the work of practical work no. 2 using the DOM API (Student Monitoring project).
أعد عمل العمل التطبيقي رقم 2 باستخدام DOM API (مشروع متابعة الطلاب).

Addtional questions (optional) أسئلة إضافية (اختيارية)

Use DOM API to manipulate the Quran-juza-amma.xml جزء عمّ file.
طبّق واجهة DOM البرمجية على مستند القرآن الكريم

Optional Challenge Repeat any three of the previous exercises using the DOM API instead of XQuery. Write a brief report (one or two pages) comparing the two approaches in terms of:

- Simplicity of implementation.
- Code readability.
- Number of lines of code.
- Advantages and limitations of each approach.

تحدٍ اختياري أعد تنفيذ أي ثلاثة من التمارين السابقة باستعمال واجهة DOM API بدلاً من XQuery.
ثم اكتب تقريرًا مختصرًا (صفحة أو صفحتين) تقارن فيه بين الطريقتين من حيث:

- سهولة التنفيذ.
- سهولة قراءة الشيفرة.
- عدد أسطر الشيفرة.
- مزايا وعيوب كل طريقة.

References مراجع

- Quran Metadata from Tanzil project: `https://tanzil.net/docs/quran_metadata`
- Python xml.dom — The Document Object Model API: `https://docs.python.org/3/library/xml.dom.html`

Additional exercises تمارين للتعمق

We want to learn how to use remote APIs offered by websites in XML or JSON like:
نريد أن نتعلم كيفية استخدام واجهات برمجة التطبيقات API البعيدة التي تقدمها مواقع الويب بتنسيق XML أو JSON مثل:

- Meteo : `https://openweathermap.org/`
- geographical map : `https://www.openstreetmap.org`
- FootBall: `https://rapidapi.com/api-sports/api/api-football/`
- More API on `https://rapidapi.com`

We want to generate technical sheets from the API results, choose a theme:
نريد إنشاء بطاقات فنية من نتائج واجهة برمجة التطبيقات (API)، اختر موضوعًا:

- دليل سياحي (وصف المكان، الخريطة، الطقس، أوقات الصلاة، الخ)

- فريق كرة القدم (وصف، الصور، النتائج....)
- مقارنة الأسعار بين المواقع
- فهرس الأفلام

- A tourist brochure (description of the place, Map, Weather, Prayer times, etc.)
- Football team (description, images, results....)
- Price comparison between sites
- Movie index.
- etc...

3. Tests فحوص

Tests topics مواضيع الفحوص

Test n°1 : for chapter 1 الفحوص رقم 1 للفصل الاول

Topic No. 1

1. List the differences between DTD and XSchema
2. List the differences between using XSL/T and the DOM API
3. What are the basic rules for writing an XML document?
4. What is the difference between CDATA and PCDATA?
5. What is XML namespace?
6. Are spaces ignored by XML?
7. Can we use graphics in XML?

1. اذكر الفروق بين DTD وXschema
2. اذكر الفروق بين استخدام XSL/T وDOM API
3. ما هي القواعد الأساسية لكتابة مستند XML؟
4. ما هو الفرق بين CDATA وPCDATA؟
5. ما هي مساحة اسم name space في XML؟
6. هل يتم تجاهل المسافات (الفراغات) بواسطة XML؟
7. هل يمكن استخدام رسومات في XML؟

Topic No. 2

Project : We want to develop a server dedicated to prayer timesمواقيت الصلاة around the world, it offers an API for various applications. We assume to have an XML database of cities and their schedules during the solar year.

تحتسب **مواقيت الصلاة** للمسلمين لمدينة ما حسب خطوط الطول والعرض والارتفا ع عن سطح الأرض وتاريخ اليوم الشمسي، وعادة ما تكون ثابتة بالنسبة ليوم ما في السنة الشمسية. تحتسب مواقيت الصلاة بطرائق مختلفة لاسيما صلاتي الفجر والعشاء. يختار كل مستخدم طريقة حسب المذهب الذي يتبعه أو الدولة التي ينتمي إليها.

للمشروع أعلاه

1. أنشئ ملف تعريف DTD لمواقيت الصلاة.
2. استخدم XSLT لعرض أوقات الصلاة لمدينة معينة في تاريخ معين.
3. اكتب دالة بايثون تعرض أوقات الصلاة لمدينة معينة في تاريخ معين، من ملف XML باستخدام DOM

For the project :

1. Define a DTD for prayer times.
2. Use XSLT to display prayer times for a given city on a given date.
3. Write a python function that returns the prayer times of a given city on a given date, from an XML file using DOM.

Topic No. 3

التمرين 1 (نقطتان [0.5، 0.25*4؛ 0.5)

1. اذكر مزايا وعيوب DOM API؟
2. أعط الاسم الكامل لاختصارات الآتية ووصفا مختصرًا لها: XSD DTD DOM SGML
3. ما الفرق بين HTML وXHTML؟

التمرين 2 (3 نقاط: 1، 1، 1)
نعتبر مستندًا قرآنيًا فيه آيات، في كل آية على مجموعة من الترجمات والروابط الصوتية.

- س1. اقترح DTD لهذا المستند.

اختر سؤالين:

- س2- اكتب قطعة الكود XSLT (القالب) الذي يسمح لك بعرض الآيات باللغة الإنجليزية.
- س3- اكتب قطعة الكود DOM في لغة بايثون يعرض قائمة السور ذات أكثر من 100 آية.
- س4- اكتب قطعة الكود XSLT الذي بعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س5- اكتب جزءًا من كود DOM بلغة بايثون والذي يعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س6- حوّل DTD إلى XSD.

Exercise 1 (**2pts [0.5, 0.25*4; 0.5**)

1. List the advantages and disadvantages of DOM API?
2. Give the full name of each abbreviation and a brief definition: XSD DTD DOM SGML

3. What is the difference between HTML and XHTML?

Exercise 2 (3pts: 1, 1, 1)
We have a Quranic document where each ayat contains a set of translations and audio links.

- Q1. Propose a DTD for this document.

Choose two questions:

- Q2- Write the XSLT code fragment (template) which allows you to display the ayates in English.
- Q3- Write the DOM code fragment in python which allows you to display the list of surates which contains more than 100 ayates.
- Q4- Write the XSLT code fragment (template) which allows you to display an index ordered according to the first word of the ayates in Arabic.
- Q5- Write the fragment of the DOM code in python which allows you to display an index ordered according to the first word of the ayates in Arabic.
- Q6- Convert DTD to XSD.

Topic No. 4

لجمع مقترحات مشاريع التخرج لطلاب السنة الثانية من برنامج الماستر، تُخزَّن البيانات في ملف CSV.
يحتوي كل موضوع على عنوان، واسم مشرف، وملخص، وأهداف، ومراجع. يُخصَّص الموضوع لأحد التخصصين أو كليهما.

1. حدد تعريف نوع المستند (DTD) للموضوع.
2. اكتب برنامجًا بلغة بايثون لتحويل ملف CSV الخاص بالمواضيع إلى XML.
3. اكتب برنامجًا بلغة بايثون (باستخدام DOM) ينجز الاستعلامات التالية:
   - 1. قائمة المواضيع لكل مشرف.
   - 2. المشرف الذي اقترح أكبر عدد من المواضيع.

For collecting Master's 2 final-year project proposals, the data is stored in a CSV file.
A proposal contains a title, a supervisor name, an abstract, objectives, and references. A topic is intended for one or both specializations.

1. Define a DTD for the proposal.
2. Write a Python program that converts the CSV file of proposals into XML.
3. Write a Python program (using the DOM) that responds to the following queries:
   (a) the list of proposals per supervisor.
   (b) the supervisor who proposed the most proposals.

Test solutions حلول الفحوص

Solution No. 1

1. List the differences between DTD and XSchema

DTD	Xschema
Special syntax.	XML syntax.
Special language.	XML dialect.
Simple data.	Allows you to define complex data structures.
Simple poor typing	Allows you to control the different complex types.

2. List the differences between using XSL/T and the DOM API

XSL/T	DOM
Operating system independent	Operating system independent
Programming language independent	Programming language independent
Requires browser support	Programmable
Complex syntax	Flexible

3. What are the basic rules for writing an XML document? (closing tags, root element, case sensitive, ignoring space, nesting, attribute value in quotes)

4. What is the difference between CDATA and PCDATA?

   PCDATA (parsed data) (CDATA not parsee data).

5. What is XML namespace?

   Qualified name associated with DTD/Schema location.

6. Are spaces ignored by XML? Yes.

7. Can we use graphics in XML? Yes, inserted by link or represented in SVG.

1. اذكر الفروق بين DTD وXschema

DTD	Xschema
صياغة خاصة	صياغة بلغة XML.
لغة خاصة	لهجة XML
بيانات بسيطة	يمكن تعريف هياكل بيانات معقّدة.
أنواع بسيطة مفتقرة	يمكن تعريف أنواع معقّدة.

2. اذكر الفروق بين استخدام XSL/T وDOM API

XSL/T	DOM
مستقل عن نظام التشغيل	مستقل عن نظام التشغيل
مستقل عن لغة البرمجة	مستقل عن لغة البرمجة
يتطلب دعم المتصفح	قابل للبرمجة
صياغة معقّدة	مرونة

3. ما هي القواعد الأساسية لكتابة مستند XML؟ (إغلاق الوسوم، عنصر الجذر، حساس لحالة الأحرف، تجاهل المسافة، التراكب، قيمة الخاصية بين علامتي الاقتباس)

4. ما هو الفرق بين CDATA وPCDATA؟ PCDATA (بيانات قابلة للمعالجة) (CDATA بيانات غير قابلة للمعالجة).

5. ما هي مساحة اسم name space في XML؟ اسم مؤهّل مرتبط بمكان أو موقع ملف تعريف البيانات DTD/Schema.

6. هل يتم تجاهل المسافات (الفراغات) بواسطة XML؟
نعم

7. هل يمكن استخدام رسومات في XML؟ نعم، بواسطة رابط أو بصيغة الرسومات الشعاعية SVG.

Solution No. 2

Project : We want to develop a server dedicated to prayer timesمواقيت الصلاة around the world, it offers an API for various applications. We assume to have an XML database of cities and their schedules during the solar year.

تحتسب **مواقيت الصلاة** للمسلمين لمدينة ما حسب خطوط الطول والعرض والارتفا ع عن سطح الأرض وتاريخ اليوم الشمسي، وعادة ما تكون ثابتة بالنسبة ليوم ما في السنة الشمسية. تحتسب مواقيت الصلاة بطرائق مختلفة لاسيما صلاتي الفجر والعشاء. يختار كل مستخدم طريقة حسب المذهب الذي يتبعه أو الدولة التي ينتمي إليها.

للمشروع أعلاه

1. أنشئ ملف تعريف DTD لمواقيت الصلاة.
2. استخدم XSLT لعرض أوقات الصلاة لمدينة معينة في تاريخ معين.
3. اكتب دالة بايثون تعرض أوقات الصلاة لمدينة معينة في تاريخ معين، من ملف XML باستخدام DOM

For the project :

1. Define a DTD for prayer times.

```
<!ELEMENT PrayerTimes (city)+>
<!ATTLIST PrayerTimes method #CDATA>
<!ELEMENT city (day)+>
<!ATTLIST city name #CDATA
 latitude #CDATA
 longitude #CDATA>
<!ELEMENT day (date, fadjr, dohr, asr, magrib, ishaa)>
<!ELEMENT date #PCDATA>
<!ELEMENT fadjr #PCDATA>
<!ELEMENT dohr#PCDATA>
<!ELEMENT asr #PCDATA>
<!ELEMENT magrib #PCDATA>
<!ELEMENT ishaa #PCDATA>
```

2. Use XSLT to display prayer times for a given city on a given date.

```
<xsl:template match="/city">

<xsl:for-each select="city[@name='Bouira']">
<h1 align="center"></h1>

<table border=1 align="center">
<xsl:for-each select="day">
    <xsl:test select="day/date == '15/12/2021'">
<tr>
    <td> Fadjr </td> <td> <xsl:value-of select="fadjr"/> </td>
</tr>
```

```
<tr>
    <td> Dohr </td> <td> <xsl:value-of select="dohr"/> </td>
</tr>
<tr>
    <td> Asr </td> <td> <xsl:value-of select="asr"/> </td>
</tr>
<tr>
    <td> Maghrib </td> <td> <xsl:value-of select="maghrib"/> </td>
</tr>
<tr>
    <td> Ishaa </td> <td> <xsl:value-of select="ishaa"/> </td>
</tr>

</xsl:for-each>
</xsl:test>


</table>
</xsl:for-each>
</xsl:template>
```

3. Write a python function that returns the prayer times of a given city on a given date, from an XML file using DOM.

```
<xsl:template match="/city">

<xsl:for-each select="city[@name='Bouira']">
<h1 align="center"></h1>

<table border=1 align="center">
<xsl:for-each select="day">
    <xsl:test select="day/date == '15/12/2021'">
<tr>
    <td> Fadjr </td> <td> <xsl:value-of select="fadjr"/> </td>
</tr>
<tr>
    <td> Dohr </td> <td> <xsl:value-of select="dohr"/> </td>
</tr>
<tr>
    <td> Asr </td> <td> <xsl:value-of select="asr"/> </td>
</tr>
<tr>
    <td> Maghrib </td> <td> <xsl:value-of select="maghrib"/> </td>
</tr>
<tr>
    <td> Ishaa </td> <td> <xsl:value-of select="ishaa"/> </td>
</tr>

</xsl:for-each>
</xsl:test>


</table>
</xsl:for-each>
```

```
</xsl:template>
```

Solution No. 3

التمرين 1 (نقطتان [0.5، 0.25*4؛ 0.5)

1. اذكر مزايا وعيوب DOM API؟
2. أعط الاسم الكامل لاختصارات الآتية ووصفا مختصرًا لها: XSD DTD DOM SGML
3. ما الفرق بين HTML وXHTML؟

التمرين 2 (3 نقاط: 1، 1، 1)
نعتبر مستندًا قرآنيًا فيه آيات، في كل آية على مجموعة من الترجمات والروابط الصوتية.

- س1. اقترح DTD لهذا المستند.

اختر سؤالين:

- س2- اكتب قطعة الكود XSLT (القالب) الذي يسمح لك بعرض الآيات باللغة الإنجليزية.
- س3- اكتب قطعة الكود DOM في لغة بايثون يعرض قائمة السور ذات أكثر من 100 آية.
- س4- اكتب قطعة الكود XSLT الذي بعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س5- اكتب جزءًا من كود DOM بلغة بايثون والذي يعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س6- حوّل DTD إلى XSD.

Exercise 1 (2pts [0.5, 0.25*4; 0.5)

1. List the advantages and disadvantages of DOM API? advantages: مزايا
   - DOM XML is language and platform independent مستقل عن لغات البرمجة ومنصات التشغيل
   - XML DOM is browsed قابل للتنقل عبره
   - XML DOM is editable قابل للتعديل

   Disadvantages: مساوئ
   - Consumes memory: if the XML structure is large) as a program written once remains in memory all the time until and unless explicitly removed.
     يستهلك كثيرًا من الذاكرة لاسيما إن كان حجم ملف البيانات كبيرًا.
2. Give the full name of each abbreviation and a brief definition: XSD DTD DOM SGML
   - **XSD** (XML schema Definition) : is an XML document format description language for defining the structure and content type of an XML document. لغة توصيف أنواع البيانات المحتواة في مستند XML، تستعمل نفس صياغة XML في تعريف البيانات
   - **DTD** (Document Type definition): describes the grammar of the document — list of elements (or tags), attributes, their content and their arrangement. لغة توصيف أنواع البيانات المحتواة في مستند XML
   - **DOM** (Document Object model): is an application programming interface (API) for well-formed XML documents. . XML واجهة برمجية للتطبيقات، لمعالجة ملفات
   - **SGML** (standard generalized Markup language) is a markup description language, ISO standard. ISO لغة توصيف بيانات عامّة، معيار تابع لمنظمة المعايير الدولية

3. What is the difference between HTML and XHTML?

   - Flexible framework requiring lenient HTML specific parser.
   - Restrictive subset of XML and needs to be parsed with standard XML parsers.

XHTML هو إطار عمل مرن يتطلب محلل HTML صارمًا. مجموعة فرعية مقيدة من XML ويجب تحليلها باستخدام محللات XML القياسية.

Exercise 2 (3pts: 1, 1, 1)
We have a Quranic document where each ayat contains a set of translations and audio links.

- Q1. Propose a DTD for this document.

```
<!ELEMENT mushaf (sura)+>
<!ELEMENT sura (aya)+>
<!ATTLIST sura
id CDATA #REQUIRED
name CDATA #REQUIRED
nb_aya CDATA #REQUIRED>
<!ELEMENT aya (arabic, translation+, audio+)>
<!ATTLIST aya id CDATA #REQUIRED>
<!ELEMENT arabic PCDATA>
<!ELEMENT translation PCDATA>
<!ATTLIST translation lang CDATA #REQUIRED>
<!ATTLIST audio reciter CDATA #REQUIRED>
```

Choose two questions:

- Q2- Write the XSLT code fragment (template) which allows you to display the ayates in English.

```
<xsl:for-each select="surat/aya/translation[@lang='english']">
    ...
</xsl:for-each>
```

- Q3- Write the DOM code fragment in python which allows you to display the list of surates which contains more than 100 ayates.

```
result =[]
for srt in xmldoc.getElementsByTagName('sura'):
     nb_aya = srt.getAttribute('nb_aya')
     name = srt.getAttribute('name')
     if  nb_aya >100 :
       result.append({"aya":aya_id, "sura":name, "aya_text":aya_text})
print(result)
```

- Q4- Write the XSLT code fragment (template) which allows you to display an index ordered according to the first word of the ayates in Arabic.

```
<xsl:apply-templates select="mushaf">
    <xsl:sort select="*/*/aya/arabic"/>
</xsl:apply-templates>
```

- Q5- Write the fragment of the DOM code in python which allows you to display an index ordered according to the first word of the ayates in Arabic.

```
result =[]
for aya in xmldoc.getElementsByTagName('arabic'):
     result =  aya.firstChild.data
result = sorted(result)
print(result)
```

- Q6- Convert DTD to XSD.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="translation">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="lang"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="audio">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="qari"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="aya">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="translation" maxOccurs="unbounded" minOccurs="1"/>
        <xs:element ref="audio" maxOccurs="unbounded" minOccurs="1"/>
      </xs:sequence>
      <xs:attribute type="xs:interger" name="id"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="sura">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="aya" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute type="xs:integer" name="nb_aya" use="optional"/>
      <xs:attribute type="xs:integer" name="id" use="optional"/>
      <xs:attribute type="xs:string" name="name" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="mushaf">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="sura" maxOccurs="114" minOccurs="1"/>
```

```
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Solution No. 4

لجمع مقترحات مشاريع التخرج لطلاب السنة الثانية من برنامج الماستر، تُخزَّن البيانات في ملف CSV.
يحتوي كل موضوع على عنوان، واسم مشرف، وملخص، وأهداف، ومراجع. يُخصَّص الموضوع لأحد التخصصين أو كليهما.

1. حدد تعريف نوع المستند (DTD) للموضوع.
2. اكتب برنامجًا بلغة بايثون لتحويل ملف CSV الخاص بالمواضيع إلى XML.
3. اكتب برنامجًا بلغة بايثون (باستخدام DOM) ينجز الاستعلامات التالية:

- 1. قائمة المواضيع لكل مشرف.
- 2. المشرف الذي اقترح أكبر عدد من المواضيع.

For collecting Master's 2 final-year project proposals, the data is stored in a CSV file.
A proposal contains a title, a supervisor name, an abstract, objectives, and references. A topic is intended for one or both specializations.

1. Define a DTD for the proposal.
2. Write a Python program that converts the CSV file of proposals into XML.
3. Write a Python program (using the DOM) that responds to the following queries:
   (a) the list of proposals per supervisor.
   (b) the supervisor who proposed the most proposals.

DTD file

```
<!ELEMENT Themes (theme+)>
<!ELEMENT Theme (title, supervisor,co-supervisor, speciality, abstract,
 objective+, reference+)>
<!ELEMENT title #PCDATA>
<!ELEMENT supervisor #PCDATA>
<!ELEMENT speciality #PCDATA>
<!ELEMENT abstract #PCDATA>
<!ELEMENT objective #PCDATA>
<!ELEMENT reference #PCDATA>
```

Python DOM script

```
<!ELEMENT mushaf (sura)+>
<!ELEMENT sura (aya)+>
<!ATTLIST sura
id CDATA #REQUIRED
name CDATA #REQUIRED
nb_aya CDATA #REQUIRED>
<!ELEMENT aya (arabic, translation+, audio+)>
```

```
<!ATTLIST aya id CDATA #REQUIRED>
<!ELEMENT arabic PCDATA>
<!ELEMENT translation PCDATA>
<!ATTLIST translation lang CDATA #REQUIRED>
<!ATTLIST audio reciter CDATA #REQUIRED>
```

4. Exams امتحانات

Topics مواضيع

Topic No. 1

مشروع :

بمناسبة كأس إفريقيا للأمم للمحليين المنظمة في الجزائر 2023، يمكن حجز تذاكر المباريات من موقع "تذكرتي"، الدفع بالبطاقة الذهبية أو البنكية، تستلم وصل دفع، ثم تذهب إلى أقرب مركز لطباعة التذكرة،

هذه صورة لحساب مستخدم في موقع "تذكرتي" :

National identification number :	160345206800000007		
Username :	أحمد بن أحمد	Address :	Bouira
First name :	أحمد	Gender :	Male
Last name :	بن أحمد	Phone :	(+213) 066-666-666
Email :	wefawe6521@prolug.com		
Date of Birth :	October 15, 2002		

تمرين 1 (20 نقاط :4، 4 ، 4*3) : تطبيقي

نريد حفظ بيانات المستخدم في مشروع "تذكرتي" في ملف XML، ثم معالجته.

1. 1. أنشئ تعريفا لهيكل البياناتXchema (وحدّد قيود أنواع البريد الإلكتروني، والجنس ورقم الهاتف)
2. استعمل xslt أو Python DOM لعرض البيانات كما في الصورة أعلاه
3. استعمل python dom لعرض بعض الإحصائيات :

(a) توزيع نسبة الذكور والإناث

(b) توزيع الأعمار كل 10 سنوات

(c) توزيع المتفرجين حسب الولايات.

Topic No. 2

التمرين 1 (8 نقاط: 2، 1.5*4)) أسئلة المنهجية

1. ما القواعد الأساسية لكتابة مستند XML؟

2. ما الفرق بين CDATA و PCDATA؟
3. ما هو نطاق الاسم name space للغة XML؟
4. هل يتم تجاهل المسافات بواسطة XML؟
5. هل يمكننا استخدام الرسوميات في XML؟

المشروع: (التمرين 2)

نريد تصميم نظام لإدارة الامتحانات ذات الإجابات المتعددة (MCQ أسئلة متعددة الخيارات) عن الويب.

بنك الأسئلة فيه عدد كبير من الأسئلة ذات الخيارات. كل سؤال فيه الحل الصحيح، ويمكن أن يحوي على الإجابة التي اختارها المستخدم.

التمرين 2 (12 نقاط [2*6]) XML:

أسئلة :

1. عرّف DTD للأسئلة ذات الخيارات .
2. أعط مخطط Xschema للأسئلة ذات الخيارات.
3. اعرض الأسئلة باستخدام XSLT.
4. اعرض نتيجة الاختبار باستخدام XSLT.
5. اعرض الأسئلة باستخدام Python/DOM.
6. اعرض نتيجة الاختبار والعلامة باستخدام Python/DOM.

Exercise 1 (8 pts: 2, [1,5*4]) Methodology questions

1. What are the basic rules for writing an XML document?
2. What is the difference between CDATA and PCDATA?
3. What is XML namespace?
4. Are spaces ignored by XML?
5. Can we use graphics in XML?

Project: (exercises 2)

We want to design a system for managing multiple choice questions (MCQ أسئلة متعددة الخيارات) for the Web.

A question bank contains a multitude of questions. Each question contains the correct solution, it can contain the answer chosen by the user.

Question Bank

1. What is XML
   - A. ○ eXentible Markup Language
   - B. ○ eXpandable Style Language
   - C. ○ eXtensible Stylesheet Language
   - D. ○ eXtra Style Language

Exercise 2 (10 pts [2*5]) XML:

Questions :

1. Define a DTD for MCQ.

2. Define an XSchema for MCQ.
3. Show questions with XSLT.
4. View test result with XSLT.
5. Show questions with Python/DOM.
6. Show test result and grade with Python/DOM.

Topic No. 3

مشروع

يرغب عبد الحميد -طالب في السنة النهائية- في إنشاء شركته الناشئة، فأنشأ تطبيق جوال يُدعى "حرفتي"، مع موقع إلكتروني وخادم مخصصين للبحث عن الحرفيين وحجز خدماتهم، مثل البنائين والدهّانين وغيرهم. يوفر هذا الموقع استضافة لملفات تعريف كل من الحرفيين والزبائن.

يقترح عبد الحميد نموذج عمل قائم على العمولة: نسبة مئوية من قيمة الأعمال المبرمة بين الحرفيين والزبائن.

إذا كانت الإيرادات أقل من 100,000 دينار جزائري، فلا عمولة؛ وإذا كانت الإيرادات بين 100,000 و1,000,000 دينار جزائري، فإنه يحصل على 2%؛ وإذا تجاوزت الإيرادات 1,000,000 دينار جزائري، يحصل الموقع على 3%.

بالإضافة إلى ذلك، يمكن للعملاء تقييم الحرفيين بنظام نجوم الخمس أو الإبلاغ عن أي نشاط احتيالي.

1. حدد الحقول في ملف تعريف الحرفي.
2. حدد تعريف نوع المستند (DTD) لملف تعريف الحرفي.
3. ما هي الحقول التي يمكن تعريفها بشكل أفضل باستخدام Xschema؟
4. اكتب برنامجًا بلغة بايثون يستورد قائمة الحرفيين من ملف CSV إلى ملف XML.
5. اكتب برنامجًا بلغة بايثون (باستخدام DOM) ينفذّ الطلبات التالية:

(a) - قائمة الحرفيين المحتالين.

(b) - قائمة أقرب البنائين المؤهلين.

Hamid, a final-year student, wants to create his own startup, based on a mobile application called "Hirfaty" (حرفتي), with a website and a server for "searching for and booking tradespeople" such as masons, painters, etc. This site provides hosting for the profiles of both tradespeople and clients.

Hamid proposes a commission-based business model: a percentage of the business concluded between tradespeople and clients. If the revenue is less than 100,000 DA, there is no commission; if the revenue is between 100,000 and 1,000,000 DA, he takes 2%; if the revenue exceeds 1,000,000 DA, the site takes 3%.

Furthermore, clients can rate tradespeople on a 5-star scale or report fraudulent activity.

1. Define the fields in a tradespeople's profile.
2. Define the tradespeople's profile DTD.
3. Which fields can be better defined using Xschema?
4. Write a Python program that imports the list of tradespeople from a CSV file into XML.
5. Write a Python program (using the DOM) that responds to the following requests:
   (a) - The list of fraudulent tradespeople.
   (b) - The list of the nearest qualified masons.

Topic No. 4

Here you have the following XML file:

```xml
<Company>
   <Address type="DZ-Address">
      <Name>Main office</Name>
      <Street>Mohamed VI Blvd.</Street>
      <City>Sidi M'Hamed</City>
      <Wilaya>Algiers</Wilaya>
      <Zip>16000</Zip>
   </Address>

   <Division>
      <DivisionName>Sales</DivisionName>
      <Location>Oran</Location>

      <Person Manager="true" Degree="master">
         <FirstName>Ben Alim</FirstName>
         <LastName>Anter</LastName>
         <PhoneExtension>077</PhoneExtension>
         <Email>anterben@work.com</Email>
      </Person>

      <Person Manager="false" Degree="bachelor">
         <FirstName>Bakhti</FirstName>
         <LastName>Boudali</LastName>
         <PhoneExtension>066</PhoneExtension>
         <Email>boudali@work.com</Email>
      </Person>
   </Division>

   <Division>
      ...
   </Division>
</Company>
```

Answer the following questions:

1. Propose a DTD for this XML document.
2. Whenever possible, indicate whether the following constraints can be represented using a DTD and/or an XML Schema (XSD).
   (a) The value of the `Degree` attribute must be either `master` or `bachelor`.
   (b) An email address must follow the format `string.string@string`.
   (c) `DZ-Address` is an extension of `Address` that adds the elements `Wilaya` and `Zip`.
   (d) The value of the `Zip` element must be between `01000` and `48000`.
   (e) The `Person` elements in XML instances must be sorted alphabetically.
3. Write a Python program using the DOM API that displays the people belonging to a given division in a table.
4. Write a Python program using the DOM API that displays only the people whose degree is `master`.

5. Redo the two last questions using XQuery.

إليك ملف XML أعلاه. أجب عن الأسئلة التالية.

1. اقترح تعريفًا من نوع DTD لهذا المستند XML.
2. عندما يكون ذلك ممكنًا، بيّن ما إذا كان يمكن تمثيل القيود التالية باستعمال DTD و/أو XSD) XML Schema).

(a) يجب أن تكون قيمة الخاصية Degree إحدى القيمتين master أو bachelor.
(b) يجب أن يكون البريد الإلكتروني بالشكل string.string@string.
(c) يمثل DZ-Address امتدادًا للنوع Address بإضافة العنصرين Wilaya و Zip.
(d) يجب أن تكون قيمة العنصر Zip بين 01000 و48000.
(e) يجب أن تكون عناصر Person مرتبة ترتيبًا أبجديًا.

3. اكتب برنامجًا بلغة Python باستعمال واجهة DOM يعرض الأشخاص المنتمين إلى قسم (Division) معين في جدول.
4. اكتب برنامجًا بلغة Python باستعمال واجهة DOM يعرض فقط الأشخاص الحاصلين على درجة `master`.
5. أعد السؤالين الأخيرين باستعمال Xquery

Exam solutions حلول امتحانات

Solution No. 1

مشروع : examS12023sol

بمناسبة كأس إفريقيا للأمم للمحليين المنظمة في الجزائر 2023، يمكن حجز تذاكر المباريات من موقع "تذكرتي"، الدفع بالبطاقة الذهبية أو البنكية، تستلم وصل دفع، ثم تذهب إلى أقرب مركز لطباعة التذكرة،
هذه صورة لحساب مستخدم في موقع "تذكرتي" :

National identification number :	160345206800000007		
Username :	أحمد بن أحمد	Address :	Bouira
First name :	أحمد	Gender :	Male
Last name :	بن أحمد	Phone :	(+213) 066-666-666
Email :	wefawe6521@prolug.com		
Date of Birth :	October 15, 2002		

تمرين 1 (20 نقاط :4، 4 ، 4*3) : تطبيقي
نريد حفظ بيانات المستخدم في مشروع "تذكرتي" في ملف XML، ثم معالجته.

1. أنشئ تعريفا لهيكل البياناتXchema (وحدّد قيود أنواع البريد الإلكتروني، والجنس ورقم الهاتف)
2. استعمل xslt أو Python DOM لعرض البيانات كما في الصورة أعلاه
3. استعمل python dom لعرض بعض الإحصائيات :
   (a) توزيع نسبة الذكور والإناث
   (b) توزيع الأعمار كل 10 سنوات
   (c) توزيع المتفرجين حسب الولايات.

Xschema

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Created with Liquid Technologies Online Tools 1.0
(https://www.liquid-technologies.com) -->
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="users">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="user" type="userType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="userType">
    <xs:sequence>
      <xs:element name="nid" type="xs:string"/>
      <xs:element name="username" type="xs:string" />
      <xs:element name="firstname" type="xs:string" />
      <xs:element name="lastname" type="xs:string" />
```

```
      <xs:element name="gender" type="genderType" />
      <xs:element name="email" type="emailType" />
      <xs:element name="birthdate" type="xs:date" />
      <xs:element name="address" type="xs:string" />
      <xs:element name="phone" type="phoneType" />
    </xs:sequence>
  </xs:complexType>


    <xs:simpleType name="genderType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="male"/>
        <xs:enumeration value="female"/>
      </xs:restriction>

    </xs:simpleType>
  <xs:simpleType name="emailType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[^@]+@[^\.]+\..+"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="phoneType">
  <xs:restriction base="xs:string">
   <xs:pattern value="0[0-9]{9}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

XSL/T script for display

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" indent="yes"/>
<xsl:template match="/user">

<html>
<head>
<title>Student Ticket</title>
</head>

<body>
<table cellpadding="6">

<tr>
    <td colspan="4">
        <b>National identification number :</b>
        <xsl:value-of select="nid"/>
    </td>
</tr>

<tr>
    <td><b>Username :</b></td>
    <td><xsl:value-of select="username"/></td>
```

```
    <td><b>Address :</b></td>
    <td><xsl:value-of select="address"/></td>
</tr>

<tr>
    <td><b>First name :</b></td>
    <td><xsl:value-of select="firstname"/></td>

    <td><b>Gender :</b></td>
    <td><xsl:value-of select="gender"/></td>
</tr>

<tr>
    <td><b>Last name :</b></td>
    <td><xsl:value-of select="lastname"/></td>

    <td><b>Phone :</b></td>
    <td><xsl:value-of select="phone"/></td>
</tr>

<tr>
    <td><b>Email :</b></td>
    <td colspan="3">
        <xsl:value-of select="email"/>
    </td>
</tr>

<tr>
    <td><b>Date of Birth :</b></td>
    <td colspan="3">
        <xsl:value-of select="birthdate"/>
    </td>
</tr>

</table>

</body>
</html>

</xsl:template>

</xsl:stylesheet>
```

Python script for display

```
from xml.dom import minidom

doc = minidom.parse("user.xml")

def get(tag):
    return doc.getElementsByTagName(tag)[0].firstChild.nodeValue

print(f"National identification number : {get('nid')}\n")
```

```
print(f"Username   : {get('username'):20} Address : {get('address')}")
print(f"First name : {get('firstname'):20} Gender  : {get('gender')}")
print(f"Last name  : {get('lastname'):20} Phone   : {get('phone')}")
print(f"Email      : {get('email')}")
print(f"Birth date : {get('birthdate')}")
```

Python script for statistics

```
def get_rate_gender(xmldoc):
    """
    gender distribution
    """
    gender_table={"male":0,
    "female":0}
    genders = xmldoc.getElementsByTagName('gender')
    for gender in genders:
        value = gender.firstChild.data
        if value in gender_table:
            gender_table[value] +=1
    total = len(genders)
    # print rates
    for key in gender_table:
        print(key, gender_table[key], gender_table[key]/total)

def get_rate_age(xmldoc):
    """
    age distribution
    """
    age_table={0:0, # [1-10[ years
    1:0,# [10-20[ years
    2:0,# [20-30[ years
    3:0,# [30-40[ years
    4:0,# [40-50[ years
    5:0,# [50-60[ years
    }
    births = xmldoc.getElementsByTagName('birthdate')
    for birth in births:
        value = birth.firstChild.data
        year = value.split("-")[0]
        age = 2023 - int(year)
        decade = age // 10 # div 10
        if decade in age_table:
            age_table[decade] +=1
    total = len(births)
    # print rates
    for key in age_table:
        print("[%d0 -  %d0["%(key, key+1),
        age_table[key], age_table[key]/total)
def get_rate_address(xmldoc):
    """
    address distribution
    """
    address_table={
```

```
    }
    addresses = xmldoc.getElementsByTagName('address')
    for address in addresses:
        value = address.firstChild.data
        if value in address_table:
            address_table[value] +=1
        else:
            address_table[value] =1
    total = len(addresses)
    # print rates
    for key in address_table:
        print(key, address_table[key], address_table[key]/total)
DATA_FILE = 'data/tadkiratidata.xml'
import xml.dom.minidom as minidom
import sys
def main():
    try:
        # o
        xmldoc = minidom.parse(DATA_FILE)
    except:
        print("Can't Open the file", DATA_FILE)
        sys.exit()

    get_rate_gender(xmldoc)
    get_rate_age(xmldoc)
    get_rate_address(xmldoc)

    return 0
if __name__ == '__main__':
    main()
```

Solution No. 2

examS12022sol

التمرين 1 (8 نقاط: 2، 1.5*4)) أسئلة المنهجية

1. ما القواعد الأساسية لكتابة مستند XML؟
2. ما الفرق بين CDATA و PCDATA؟
3. ما هو نطاق الاسم name space للغة XML؟
4. هل يتم تجاهل المسافات بواسطة XML؟
5. هل يمكننا استخدام الرسوميات في XML؟

المشروع: (التمرين 2)

نريد تصميم نظام لإدارة الامتحانات ذات الإجابات المتعددة (MCQ أسئلة متعددة الخيارات) عن الويب.

بنك الأسئلة فيه عدد كبير من الأسئلة ذات الخيارات. كل سؤال فيه الحل الصحيح، ويمكن أن يحوي على الإجابة التي اختارها المستخدم.

التمرين 2 (12 نقاط [6*2]) XML:

أسئلة :

1. عرّف DTD للأسئلة ذات الخيارات .

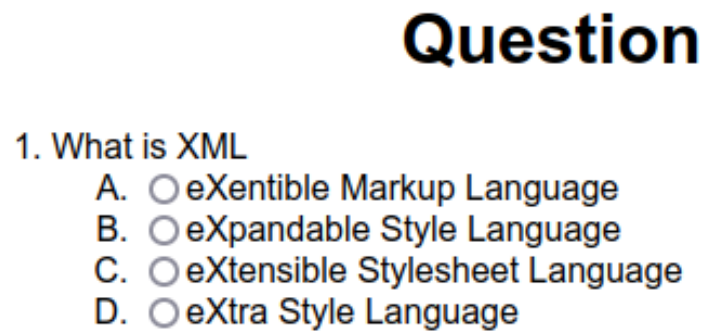
2. أعط مخطط Xschema للأسئلة ذات الخيارات.
3. اعرض الأسئلة باستخدام XSLT.
4. اعرض نتيجة الاختبار باستخدام XSLT.
5. اعرض الأسئلة باستخدام Python/DOM.
6. اعرض نتيجة الاختبار والعلامة باستخدام Python/DOM.

Exercise 1 (8 pts: 2, [1،5*4]) Methodology questions

1. What are the basic rules for writing an XML document?
   (closing tags, root element, case sensitive, ignoring space, nesting, attribute value in quotes)
2. What is the difference between CDATA and PCDATA?
   PCDATA (parsed data) (CDATA not parsee data).
3. What is XML namespace?
   Qualified name associated with DTD/Schema location.
4. Are spaces ignored by XML?
   Yes.
5. Can we use graphics in XML?
   Yes, inserted by link or represented in SVG.

Project: (exercises 2)
We want to design a system for managing multiple choice questions (MCQ أسئلة متعددة الخيارات) for the Web.

A question bank contains a multitude of questions. Each question contains the correct solution, it can contain the answer chosen by the user.



Exercise 2 (10 pts [2*5]) XML:
Questions :

1. Define a DTD for MCQ.
   Example of XML file للتوضيح فقط، غير مطلوب

```
<questionbank>
<module>Méthodes et technologies d'implémentation'</module>
 <question id="1">
 <label>What is XML</label>
    <choice correct="true" chosen="true">
       eXentible Markup Language
    </choice>
```

```
    <choice correct="false" chosen="false">
       eXpandable Style Language
    </choice>
    <choice correct="false" chosen="false">
       eXtensible Stylesheet Language
    </choice>
    <choice correct="false" chosen="false">
       eXtra Style Language
    </choice>
 </question>
<question id="2">
 <label>Which option is true about session scope?</label>
    <choice correct="false" chosen="false">
     Objects are accessible only from the page in which they
     are created
    </choice>
    <choice correct="true" chosen="false">
    Objects are accessible only from the pages which are in same
    session
    </choice>
    <choice correct="false" chosen="false">
    Objects are accessible only from the pages
    which are processing
    the same request
    </choice>
    <choice correct="false" chosen="true">
    Objects are accessible only from the pages which reside
    in same application
    </choice>
 </question>
</questionbank>
```

DTD file

```
<!ELEMENT questionbank (module, question+)>
<!ELEMENT question (label,choice+)>
<!ATTLIST question id CDATA #REQUIRED>
<!ELEMENT label (#PCDATA)>
<!ELEMENT choice (#PCDATA)>
<!ELEMENT module (#PCDATA)>
<!ATTLIST choice correct CDATA #IMPLIED chosen CDATA #IMPLIED>
```

2. Define an XSchema for MCQ.

```
<xs:element name="questionbank">
  <xs:complexType name="QuestionBankType">
     <xs:sequence>
        <xs:element name="module" type="String" />
        <xs:element maxOccurs="unbounded" name="question"
                  type="QuestionType" />
     </xs:sequence>
  </xs:complexType>
   </xs:element>
```

```
    <xs:complexType name="QuestionType">
      <xs:sequence>
       <xs:element name="label" type="String" />
       <xs:element maxOccurs="5" name="choice" type="ChoiceType" />
      </xs:sequence>
       <xs:attribute name="id" type="Integer" use="required" />
    </xs:complexType>
    <xs:complexType name="ChoiceType">
       <xs:attribute default="false" name="correct" type="String" />
       <xs:attribute default="false" name="chosen" type="String" />
    </xs:complexType>
</xs:schema>
```

3. Show questions with XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" encoding="utf-8"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/questionbank">
<h1 align="center">
 Question Bank of
 <xsl:value-of select="module" />
</h1>
<ol>
 <xsl:for-each select="question">
    <li>
       <xsl:value-of select="label" />
       <ol type="A">
          <xsl:for-each select="choice">
             <li>
                <input type="radio" />
                <xsl:value-of select="." />
             </li>
          </xsl:for-each>
       </ol>
    </li>
    <br />
 </xsl:for-each>
</ol>
</xsl:template>
</xsl:stylesheet>
```

Example of display غير مطلوب، فقط للتوضيح

Question Bank of Méthodes et tech

1. What is XML
   - A. ○ eXentible Markup Language
   - B. ○ eXpandable Style Language
   - C. ○ eXtensible Stylesheet Language
   - D. ○ eXtra Style Language

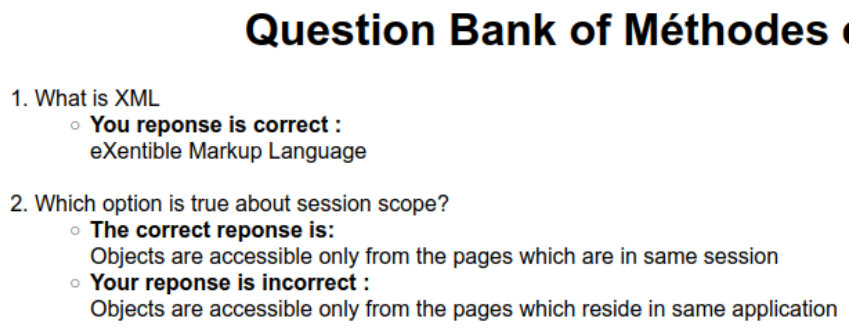
2. Which option is true about session scope?
   - A. ○ Objects are accessible only from the page in which they are created
   - B. ○ Objects are accessible only from the pages which are in same session
   - C. ○ Objects are accessible only from the pages which are processing the same request
   - D. ○ Objects are accessible only from the pages which reside in same application

4. View test result with XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" encoding="utf-8"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system=
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/questionbank">
<h1 align="center">
 Question Bank of
 <xsl:value-of select="module" />
</h1>
<ol>
 <xsl:for-each select="question">
    <li>
       <xsl:value-of select="label" />
       <ul>
          <xsl:for-each select="choice">
             <xsl:choose>
               <xsl:when
                    test="(@correct='true') and (@chosen='true')">
                 <li>
                      <strong>You reponse is correct :</strong>
                      <br />
                      <xsl:value-of select="." />
                 </li>
               </xsl:when>
               <xsl:when test="@correct='true'">
                  <li>
                     <strong>The correct reponse is:</strong>
                     <br />
                     <xsl:value-of select="." />
                  </li>
               </xsl:when>
               <xsl:when test="@chosen='true'">
                  <li>
                     <strong>Your reponse is incorrect :</strong>
                     <br />
                     <xsl:value-of select="." />
                  </li>
               </xsl:when>
             </xsl:choose>
          </xsl:for-each>
```

```
        </ul>
    </li>
    <br />
 </xsl:for-each>
</ol>
</xsl:template>
</xsl:stylesheet>
```

Example of score display للتوضيح فقط، غير مطلوب



Question Bank of Méthodes

1. What is XML
   - **You reponse is correct :**
     eXentible Markup Language

2. Which option is true about session scope?
   - **The correct reponse is:**
     Objects are accessible only from the pages which are in same session
   - **Your reponse is incorrect :**
     Objects are accessible only from the pages which reside in same application

5. Show questions with Python/DOM.
6. Show test result and grade with Python/DOM.

```
mport xml.dom.minidom as minidom
import sys

def show_question(xmldoc):
    """ Show questions """
    module = xmldoc.getElementsByTagName("module")[0].firstChild.data
    questions = xmldoc.getElementsByTagName("question")
    for question in questions:
        qid = question.getAttribute("id")
        label = question.getElementsByTagName("label")[0].firstChild.data
        choices = question.getElementsByTagName("choice")
        print(qid,"-",label)
        for i,choice in enumerate(choices):
            reponse = choice.firstChild.data
            print("\t", "%d-"%i, reponse)
def show_responses(xmldoc):
    """ Show responses with score """
    score = 0
    module = xmldoc.getElementsByTagName("module")[0].firstChild.data
    questions = xmldoc.getElementsByTagName("question")
    for question in questions:
        qid = question.getAttribute("id")
        label = question.getElementsByTagName("label")[0].firstChild.data
        choices = question.getElementsByTagName("choice")
        print(qid,"-",label)
        # show responses
        for i,choice in enumerate(choices):
            reponse = choice.firstChild.data
            correct = choice.getAttribute("correct")
            chosen = choice.getAttribute("chosen")
```

```
            if correct=="true" and chosen=="true":
                print('Your response is correct')
                print("\t", "%d-"%i, reponse)
                score +=1
            elif correct=="true":
                print('The correct answer is ')
                print("\t", i, "-", reponse)
            elif chosen=="true":
                print('Your response is ')
                print("\t", "%d-"%i, reponse)
        print("Score is ", score)
if __name__ == "__main__":
    datafile = "qcm.xml"
    try:
        xmldoc = minidom.parse(datafile)
    except:
        print("Can't open file", datafile)
        sys.exit()
    show_question(xmldoc)
    show_responses(xmldoc)
```

Example of display للتوضيح فقط، غير مطلوب

```
1- What is XML
   0- eXentible Markup Language
   1- eXpandable Style Language
   2- eXtensible Stylesheet Language
   3- eXtra Style Language
2- Which option is true about session scope?
   0- Objects are accessible only from the page in which they
       are created
   1- Objects are accessible only from the pages which are in same
    session
   2- Objects are accessible only from the pages which are processing
    the same request
   3- Objects are accessible only from the pages which reside in same
    application
```

Example of score display للتوضيح فقط، غير مطلوب

```
1- What is XML
   0- eXentible Markup Language
   1- eXpandable Style Language
   2- eXtensible Stylesheet Language
   3- eXtra Style Language
2- Which option is true about session scope?
   0- Objects are accessible only from the page in which they
       are created
   1- Objects are accessible only from the pages which are in same
    session
   2- Objects are accessible only from the pages which are processing
    the same request
```

```
3- Objects are accessible only from the pages which reside in same
 application
```

Solution No. 3

examS12021sol

مشروع

يرغب عبد الحميد -طالب في السنة النهائية- في إنشاء شركته الناشئة، فأنشأ تطبيق جوال يُدعى "حرفتي"، مع موقع إلكتروني وخادم مخصصين للبحث عن الحرفيين وحجز خدماتهم، مثل البنائين والدهّانين وغيرهم. يوفر هذا الموقع استضافة لملفات تعريف كل من الحرفيين والزبائن.

يقترح عبد الحميد نموذج عمل قائم على العمولة: نسبة مئوية من قيمة الأعمال المبرمة بين الحرفيين والزبائن.

إذا كانت الإيرادات أقل من 100,000 دينار جزائري، فلا عمولة؛ وإذا كانت الإيرادات بين 100,000 و1,000,000 دينار جزائري، فإنه يحصل على 2%؛ وإذا تجاوزت الإيرادات 1,000,000 دينار جزائري، يحصل الموقع على 3%.

بالإضافة إلى ذلك، يمكن للعملاء تقييم الحرفيين بنظام نجوم الخمس أو الإبلاغ عن أي نشاط احتيالي.

1. حدد الحقول في ملف تعريف الحرفي.

   يجب أن يحتوي ملف الحرفي على الأقسام التالية:

   - - المعلومات الشخصية: الاسم، اللقب، تاريخ الميلاد.
   - - معلومات الاتصال: البريد الإلكتروني، رقم الهاتف، العنوان.
   - - المعلومات المالية: الحساب البنكي، رقم الأعمال (أو الإيرادات السنوية).
   - - المعلومات المهنية: المهارات، الشهادات، مجال العمل (التخصص أو النوع).
   - - معلومات التطبيق: حالة الحساب (نشط، موقوف، مُبلّغ عنه)، وتقييم الحرفي.

2. حدد تعريف نوع المستند (DTD) لملف تعريف الحرفي. (انظر ملف DTD أسفله)

3. ما هي الحقول التي يمكن تعريفها بشكل أفضل باستخدام Xschema؟ يوفر XSD (XML Schema) إمكانيات تحقق من صحة البيانات وتحديد أنواعها بشكل أدق من DTD. وهو مناسب خصوصًا لتعريف:

   -

   - التواريخ.
   - - القيم الرقمية (الأعداد الصحيحة والعشرية).
   - - القيم المعدودة (Enumerations).
   - - السلاسل النصية ذات الصيغ المحددة أو القيود الخاصة.

   يمكن تحسين تعريف الحقول التالية باستعمال XSD: (انظر الجدول أسفله في النسخة الإنجليزية)

4. اكتب برنامجًا بلغة بايثون يستورد قائمة الحرفيين من ملف CSV إلى ملف XML.
   (انظر ملف بيثون أسفله)

5. اكتب برنامجًا بلغة بايثون (باستخدام DOM) ينفذّ الطلبات التالية:

(a) - قائمة الحرفيين المحتالين.

(b) - قائمة أقرب البنائين المؤهلين.

(انظر ملف بيثون أسفله)

Hamid, a final-year student, wants to create his own startup, based on a mobile application called "Hirfaty" (حرفتي), with a website and a server for "searching for and booking tradespeople" such as masons, painters, etc. This site provides hosting for the profiles of both tradespeople and clients.

Hamid proposes a commission-based business model: a percentage of the business concluded between tradespeople and clients. If the revenue is less than 100,000 DA, there is no commission; if the revenue is between 100,000 and 1,000,000 DA, he takes 2%; if the revenue exceeds 1,000,000 DA, the site takes 3%.

Furthermore, clients can rate tradespeople on a 5-star scale or report fraudulent activity.

1. Define the fields in a tradespeople's profile. The profile should contain the following sections:

   - Personal information: first name, last name, date of birth.
   - Contact information: email, phone number, address.
   - Financial information: bank account, annual revenue.
   - Professional information: skills, qualifications (degrees), field of work (domain/type).
   - Application information: account status (active, suspended, reported), evaluation score.

Field	Type	Category
Last Name	Text	Personal Information
First Name	Text	Personal Information
Date of Birth	Date	Personal Information
Address	Text	Contact Information
Email	Text	Contact Information
Phone Number	Text	Contact Information
Domain	Text	Professional Information
Evaluation	Integer	Application Information
Status	(Reported, Active)	Application Information
Qualification	Text	Professional Information
Annual Revenue	Real	Financial Information
Bank Account	Text	Financial Information

2. Define the tradespeople's profile DTD.

   (cf. dtd code below)

3. Which fields can be better defined using Xschema?

   XML Schema (XSD) provides richer data typing and validation than DTD. It is particularly useful for defining:

   - Dates
   - Numeric values (integers and decimals)
   - Enumerations
   - Formatted strings and pattern restrictions

   The following fields are better declared using XSD:

Field	XSD Type / Restriction
Date of Birth	`xs:date`
Email	Formatted string (`xs:string` with a pattern)
Phone Number	Formatted string (`xs:string` with a pattern)
Evaluation	`xs:integer` or `xs:decimal`
Bank Account	Formatted string (`xs:string` with a pattern)
Status	Enumeration (`Reported`, `Active`, `Suspended`)
Annual Revenue	`xs:decimal`

4. Write a Python program that imports the list of tradespeople from a CSV file into XML.
   (cf. python code below)
5. Write a Python program (using the DOM) that responds to the following requests:
   (a) - The list of fraudulent tradespeople.
   (b) - The list of the nearest qualified masons.

   (cf. python code below)

DTD file

```
<!ELEMENT artisans (artisan*)>

<!ELEMENT artisan (
    id,
    LastName,
    FirstName,
    DateOfBirth,
    Address,
    Email,
    PhoneNumber,
    Domain,
    Evaluation,
    BankAccount,
    Status,
    Qualification,
    AnnualRevenue
)>

<!ELEMENT LastName (#PCDATA)>
<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT DateOfBirth (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
<!ELEMENT PhoneNumber (#PCDATA)>
<!ELEMENT Domain (#PCDATA)>
<!ELEMENT Evaluation (#PCDATA)>
<!ELEMENT BankAccount (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT Qualification (#PCDATA)>
<!ELEMENT AnnualRevenue (#PCDATA)>
```

Python file

```
import sys
import xml.dom.minidom as minidom


def convert(filename):
    """Convert a tab-separated file into an XML artisan database."""
    try:
        fl = open(filename, encoding="utf-8")
    except OSError:
```

```
        print("Cannot open file:", filename)
        sys.exit()

    print("""<?xml version="1.0" encoding="UTF-8"?>
<artisans>""")

    for line in fl:
        fields = line.strip().split("\t")

        last_name = fields[0]
        first_name = fields[1]
        address = fields[2]
        domain = fields[3]
        status = fields[4]

        # Optional fields (if available)
        birth_date = ""
        email = ""
        phone = ""
        qualification = ""
        evaluation = "0"
        revenue = "0"
        bank_account = ""

        print(f"""
    <artisan>
        <last_name>{last_name}</last_name>
        <first_name>{first_name}</first_name>
        <birth_date>{birth_date}</birth_date>
        <address>{address}</address>
        <email>{email}</email>
        <phone>{phone}</phone>
        <domain>{domain}</domain>
        <qualification>{qualification}</qualification>
        <evaluation>{evaluation}</evaluation>
        <annual_revenue>{revenue}</annual_revenue>
        <bank_account>{bank_account}</bank_account>
        <status>{status}</status>
    </artisan>""")

    print("</artisans>")


def show_nearby_builders(xmldoc, city):
    """Display builders located in a given city."""
    artisans = xmldoc.getElementsByTagName("artisan")

    for art in artisans:
        last_name = art.getElementsByTagName(
            "last_name")[0].firstChild.data
        domain = art.getElementsByTagName(
            "domain")[0].firstChild.data
        address = art.getElementsByTagName(
            "address")[0].firstChild.data
```

```
        if domain.lower() == "builder" and address.lower() == city.lower():
            print(last_name)


def show_reported_artisans(xmldoc):
    """Display artisans whose status is 'reported'."""
    artisans = xmldoc.getElementsByTagName("artisan")

    for art in artisans:
        last_name = art.getElementsByTagName(
            "last_name")[0].firstChild.data
        status = art.getElementsByTagName(
            "status")[0].firstChild.data

        if status.lower() == "reported":
            print(f"{last_name} is a reported artisan.")


def read_xml(filename):
    """Read and process an XML artisan database."""
    try:
        xmldoc = minidom.parse(filename)
    except OSError:
        print("Cannot open XML file:", filename)
        sys.exit()

    show_reported_artisans(xmldoc)
    show_nearby_builders(xmldoc, "Bouira")
```

Solution No. 4

examS1r2020sol

Here you have the following XML file:

```
<Company>
   <Address type="DZ-Address">
      <Name>Main office</Name>
      <Street>Mohamed VI Blvd.</Street>
      <City>Sidi M'Hamed</City>
      <Wilaya>Algiers</Wilaya>
      <Zip>16000</Zip>
   </Address>

   <Division>
      <DivisionName>Sales</DivisionName>
      <Location>Oran</Location>

      <Person Manager="true" Degree="master">
         <FirstName>Ben Alim</FirstName>
         <LastName>Anter</LastName>
         <PhoneExtension>077</PhoneExtension>
         <Email>anterben@work.com</Email>
      </Person>
```

```
      <Person Manager="false" Degree="bachelor">
         <FirstName>Bakhti</FirstName>
         <LastName>Boudali</LastName>
         <PhoneExtension>066</PhoneExtension>
         <Email>boudali@work.com</Email>
      </Person>
   </Division>

   <Division>
      ...
   </Division>
</Company>
```

Answer the following questions:

1. Propose a DTD for this XML document.

```
<!ELEMENT Company (Address, Division+)>

<!ELEMENT Address (Name, Street, City, Wilaya, Zip)>
<!ATTLIST Address
          type CDATA #REQUIRED>

<!ELEMENT Name (#PCDATA)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Wilaya (#PCDATA)>
<!ELEMENT Zip (#PCDATA)>

<!ELEMENT Division (DivisionName, Location, Person+)>

<!ELEMENT DivisionName (#PCDATA)>
<!ELEMENT Location (#PCDATA)>

<!ELEMENT Person (FirstName, LastName, PhoneExtension, Email)>
<!ATTLIST Person
          Manager (true | false) #REQUIRED
          Degree CDATA #REQUIRED>

<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT LastName (#PCDATA)>
<!ELEMENT PhoneExtension (#PCDATA)>
<!ELEMENT Email (#PCDATA)>
```

2. Whenever possible, indicate whether the following constraints can be represented using a DTD and/or an XML Schema (XSD).

   (a) The value of the `Degree` attribute must be either `master` or `bachelor`.
   Possible only with XML Schema, using the enumeration facet.

   (b) An email address must follow the format `string.string@string`.
   Possible only with XML Schema, using a pattern restriction on ‘xs:string‘.

(c) `DZ-Address` is an extension of `Address` that adds the elements `Wilaya` and `Zip`.
Possible only with XML Schema, using complex type extension.

(d) The value of the `Zip` element must be between `01000` and `48000`.
Possible only with XML Schema, using the minInclusive and maxInclusive facets.

(e) The `Person` elements in XML instances must be sorted alphabetically.
Not possible with either a DTD or an XML Schema. Such a constraint must be enforced by the application or through additional validation technologies (e.g., Schematron or application code).

3. Write a Python program using the DOM API that displays the people belonging to a given division in a table.
(see python code below.)

4. Write a Python program using the DOM API that displays only the people whose degree is `master`.
(see python code below.)

5. Redo the two last questions using XQuery. (see XQuery code below.)

إليك ملف XML أعلاه. أجب عن الأسئلة التالية.

1. اقترح تعريفًا من نوع DTD لهذا المستند XML.

2. عندما يكون ذلك ممكنًا، بيّن ما إذا كان يمكن تمثيل القيود التالية باستعمال DTD و/أو XSD) XML Schema).

(a) يجب أن تكون قيمة الخاصية Degree إحدى القيمتين master أو bachelor.
يمكن تمثيل هذا القيد فقط باستخدام XML Schema باستعمال enumeration.

(b) يجب أن يكون البريد الإلكتروني بالشكل string.string@string.
يمكن تمثيل هذا القيد فقط باستخدام XML Schema باستعمال pattern على النوع 'xs:string'.

(c) يمثل DZ-Address امتدادًا للنوع Address بإضافة العنصرين Wilaya و Zip.
يمكن تمثيل هذا القيد فقط باستخدام XML Schema باستعمال امتداد الأنواع المركبة (Complex Type Extension).

(d) يجب أن تكون قيمة العنصر Zip بين 01000 و48000.
يمكن تمثيل هذا القيد فقط باستخدام XML Schema باستعمال القيدين minInclusive و maxInclusive.

(e) يجب أن تكون عناصر Person مرتبة ترتيبًا أبجديًا.
لا يمكن التعبير عن هذا القيد باستخدام DTD أو XML Schema، وإنما يجب التحقق منه بواسطة التطبيق أو باستخدام تقنيات تحقق إضافية مثل Schematron.

3. اكتب برنامجًا بلغة Python باستعمال واجهة DOM يعرض الأشخاص المنتمين إلى قسم (Division) معين في جدول.
(انظر كود بيثون أسفله)

4. اكتب برنامجًا بلغة Python باستعمال واجهة DOM يعرض فقط الأشخاص الحاصلين على درجة `master`.
(انظر كود بيثون أسفله)

5. أعد السؤالين الأخيرين باستعمال Xquery (انظرا استعلام XQuery أسفله)

Displays the people belonging to a given division.

```
import xml.dom.minidom as minidom

# Read the XML file
```

```
doc = minidom.parse("company.xml")

# Division to search
division_name = "Sales"

# Get all divisions
divisions = doc.getElementsByTagName("Division")

for division in divisions:
    name = division.getElementsByTagName(
                                    "DivisionName")[0].firstChild.data

    if name == division_name:
        print("-----------------------------------------")
        print("First Name\tLast Name\tDegree")
        print("-----------------------------------------")

        persons = division.getElementsByTagName("Person")

        for person in persons:
            first = person.getElementsByTagName(
                    "FirstName")[0].firstChild.data
            last = person.getElementsByTagName(
                            "LastName")[0].firstChild.data
            degree = person.getAttribute("Degree")

            print(first, "\t", last, "\t", degree)
```

Displays only the people whose degree is `master`.

```
import xml.dom.minidom as minidom

# Read the XML file
doc = minidom.parse("company.xml")

# Get all persons
persons = doc.getElementsByTagName("Person")

print("------------------------------")
print("First Name\tLast Name")
print("------------------------------")

for person in persons:

    degree = person.getAttribute("Degree")

    if degree == "master":
        first = person.getElementsByTagName(
        "FirstName")[0].firstChild.data
        last = person.getElementsByTagName(
        "LastName")[0].firstChild.data

        print(first, "\t", last)
```

People in the Sales division

```
for $p in doc("company.xml")/Company/Division
            [DivisionName="Sales"]/Person
return
concat(
    data($p/FirstName),
    " ",
    data($p/LastName),
    " - ",
    data($p/@Degree)
)
```

Output

```
  Ben Alim Anter - master
  Bakhti Boudali - bachelor
```

People with a master degree

```
for $p in doc("company.xml")//Person[@Degree="master"]
return
concat(
    data($p/FirstName),
    " ",
    data($p/LastName)
)
```

Output

```
    Ben Alim Anter
```

5. Appendices ملحقات

Appendice A: Resources ملحق أ: موارد

هذه قائمة من المراجع والموارد المفيدة لمادة "البيانات شبه المهيكلة":

Books كتب

- طه زروقي (2012). نبراس: دليل المصطلحات للشعب التقنية. جامعة البويرة.
- د. علاء طعيمة (2022a). بايثون عن طريق الأمثلة. 90 مشروع بايثون مع الكود المصدري تم حلها وشرحها. دار العربي. URL: https://dlarabic.com
- د. علاء طعيمة (Dec. 2022b). كيف تبرمج في بايثون. دار العربي. URL: https://dlarabic.com (visited on 07/22/2026)
- ليزا تاغليفيري (Mar. 20, 2025). البرمجة بلغة بايثون. تعلم البرمجة وكتابة البرامج وتنقيحها بلغة بايثون. Ed. by جميل بيلوني. Trans. by محمد بغات and عبد اللطيف إيمش. أكاديمية حسوب. URL: https://academy.hsoub.com/files/15-%D8%A7%D9%84%D8%A8%D8%B1%D9%85%D8%AC%D8%A9-%D8%A8%D9%84%D8%BA%D8%A9-%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86/ (visited on 07/22/2026)
- تشارلز راسل سيفرانس (2023). بايثون للجميع: التعامل مع البيانات باستخدام لغة بايثون 3. Trans. by منصة إلكترونيكس غو. منصة إلكترونيكس غو. URL: https://www.py4e.com/translations/AR/ar/book_ar_2023_01_01.pdf (visited on 07/22/2026)

Courses دروس

- W3C.org (2023). *W3Schools Tutorials*. URL: http://www.w3schools.com.
- W3Schools (2026). *XSLT Elements Reference*. URL: https://www.w3schools.com/xml/xsl_elementref.asp (visited on 07/23/2026)
- Taha Zerrouki (2026b). *Taha Zerrouki — YouTube Channel*. URL: https://www.youtube.com/@taha.zerrouki (visited on 07/23/2026)

Tools أدوات وبرامج

- XML Copy Editor Project (2026). *XML Copy Editor*. URL: https://xml-copy-editor.sourceforge.io (visited on 07/23/2026)

- BaseX Team (2026). *BaseX GUI – High-Performance XML Database Engine*. URL: https://basex.org/basex/gui/ (visited on 07/23/2026)

  Libraries

- Saxonica (2026). *saxonche: Official Python Package for SaxonC-HE*. version 13.0.0. URL: https://pypi.org/project/saxonche/ (visited on 07/23/2026)

- Python Software Foundation (2026). *xml.dom — The Document Object Model API*. URL: https://docs.python.org/3/library/xml.dom.html (visited on 07/23/2026)

Resources موارد

- Taha Zerrouki (2026a). *DSSBook Resources: Quran Data*. URL: https://github.com/linuxscout/DSSBook-resources/ (visited on 07/23/2026)

- Tanzil Project (2026). *Quran Metadata*. URL: https://tanzil.net/docs/quran_metadata (visited on 07/23/2026)

- David Abram (2026a). *XML Schema Cheat Sheet*. URL: https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/xsd-cheat-sheet.md (visited on 07/23/2026)

- David Abram (2026b). *XML Schema Cheat Sheet (PDF)*. URL: https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/cheat-sheet.pdf (visited on 07/23/2026)

- W3C XML Schema Working Group (2004). *XML Schema Structures Quick Reference*. URL: https://www.cheat-sheets.org/saved-copy/SchemaStructuresQR-2.pdf (visited on 07/23/2026)

Resources online for students مصادر دراسية للطلبة

- CS Bouira Team (2026). *CS Bouira: Academic Resources for Computer Science Students*. URL: http://csbouira.xyz/ (visited on 07/23/2026)

- Club Scientifique de l'ESI (CSE) (2026). *Trésor ESI: Educational Resources Library*. URL: https://tresor.cse.club/ (visited on 07/23/2026)

Appendice B: The course curriculum ملحق ب: منهاج المادة

البرنامج الرسمي المعتمد لمادة البيانات شبه المهيكلة، المعتمد في الجامعات الجزائرية، أقسام الإعلام الآلي، سنة ثالثة ليسانس، تخصص أنظمة الإعلام الآلي SI وتخصص هندسة نظم المعلومات والبرمجيات، حسب التعديل الجديد لسنة 2018-2019.

- **Bachelor TITLE:** Computer Science (SI/ISIL)
- **MATERIAL TITLE:** Semi-structured data
- **SEMESTRE:** 6
- **TEACHING UNITY:** Basic teaching unit: UEF2
- **CREDIT:** 5
- **COFFICIENT:** 3
- **Evaluation mode :** Exams, practical and continuous evaluation.
- **Evaluation method :** Exam (60%) , continuous evaluation. (40 %)
- **Recommended prior knowledge:** : A programming language

Educational objectives

The objective of this subject is to enable the student to become familiar with unstructured data structures to be used in web-oriented applications.

Content of the material:

- **Context and Issues**
  - Database Fundamentals
  - Multimedia and Document
  - Hypermedia, Internet, and the Web
  - Issues Addressed in this Course
- **Multimedia Documents and Hyperdocuments**
  - Documents
    - Introduction
    - Modeling of Specific Documents
    - Modeling of Document Classes
  - Hyperdocuments
  - Multimedia Content
- **XML Core**
  - Introduction to XML
  - Basic XML Structure
  - Namespaces
  - XML Schemas
- **XML Galaxy**
  - Paths: XPATH

        * Principles
        * Axes
        * Filters
        * Predicates
    - Stylesheets and XSL Processing
    - XML Applications: RDF, SVG, ...
    - Processing: DOM and SAX
    - Pointers: XPOINTER
    - Links: XLINK

- **XML Databases and Semi-Structured Databases**
    - Semi-Structured Data and XML
    - Query Languages
    - XML Databases

- **XQUERY and Databases**
    - XML and Semi-Structured Data
        * Semi-Structured Databases
        * Query Languages
        * XML Databases
    - XQUERY
        * XQuery Syntax
        * XQuery Expressions
        * Functions and Operations
        * Use Cases and Examples of Functions

References

1. A.B. Chaudhri, A. Rashid, and R. Zicari (2003). *XML Data Management: Native XML and XML-enabled Database Systems*. Addison-Wesley. ISBN: 9780201844528. URL: https://books.google.dz/books?id=7LNhdOeQulQC.

2. A. Michard (2000). *Xml. Langage Et Applications*. Eyrolles. ISBN: 9782212092066. URL: https://books.google.dz/books?id=uMVfAAAACAAJ

3. Georges Gardarin (2001). *Bases de données: objet et relationnel*. Paris: Eyrolles. ISBN: 978-2-212-09283-7

Glossary مسرد

مرتبة أبجديًا حسب الحرف اللاتيني

Assessment [*Évaluation*]	تقويم، فحص
Assiduity [*Assiduité*]	حضور
Assignment [*Affectation*]	إسناد
Associated [*Associé*]	مرتبط
Assume [*Supposer*]	يفترض
Attendance [*Présence*]	حضور
Attribute(class)[*Attribut (classe)*]	خاصية
Attribute [*Attribut*]	خاصية، سمة
Availability [*Disponibilité*]	توفر
Average [*Moyenne*]	معدّل
Bachelor [*Licence*]	ليسانس
Continuous [*Continue*]	مستمر، متواصل
Customer [*Client*]	زبون، عميل
Customize [*Personnaliser*]	يخصص
Default [*Par Défaut*]	تلقائيا
Delivery [*Remise*]	تسليم
Denote [*Noté*]	يرمز
Derive [*Dériver*]	يشتقّ
Design [*Conception*]	تصميم
Diagram [*Diagramme*]	مخطط
Dialect [*Dialecte*]	لهجة، لغة فرعية
Disadvantages [*Inconvénient*]	عيوب، مساوئ
Display [*Affichage*]	عرض
Documentation [*Documentation*]	توثيق
Document [*Document*]	مستند
Dynamic [*Dynamique*]	متحرّك، تفاعلي
Editable [*Éditable*]	قابل للتعديل، للكتابة للتحرير
Edition [*Édition*]	تعديل، تحرير، كتابة
Ensure [*Assurer*]	يضمن
Enumerate [*Énumérer*]	يعدّ
Execute [*Exécuter*]	تنفيذ، تشغيل
Explain [*Expliquer*]	شرح
Explicitly [*Explicitement*]	بوضوح
Facilitate [*Faciliter*]	يسهّل
Feature [*Fonctionnalité*]	ميزة، سمة
Fields [*Champs*]	حقول
Follow [*Suivant*]	تالي
Fragment [*Fragment*]	قطعة، جزء
Frame [*Cadre*]	إطار
Framework [*Framework*]	إطار عمل
Function [*Fonction*]	دالة، وظيفة
Grade [*Grade*]	علامة، درجة
GUI [*Interface Graphique*]	واجهة المستخدم الرسومية
Handle [*Manipuler*]	يعالج، يتحكم
Improve [*Améliorer*]	يحسّن
Inclusive [*Inclusive*]	متضمن
Instance [*Instance*]	فرد
Instance[*Instance*]	فرد
Instead [*Au Lieu*]	بدلا عن
Interface [*Interface*]	واجهة
Item [*Élément*]	عنصر
Mandatory [*Obligatoire*]	إجباري
Manipulation [*Manipulation*]	تعامل، معالجة
Mentoring [*Suivi*]	متابعة، مراقبة
Method (class)[*Méthode (classe)*]	دالة صنف
Nesting [*Imbrication*]	تداخل، تراكب
Notification [*Notification*]	إشعار، تنبيه
Object[*Objet*]	كائن، شيء
Optional [*Facultatif*]	اختياري
Overlapping [*Chevauchement*]	تعارض، تداخل
Parameters[*Paramètres*]	وسائط، مَعْلمات
Parser [*Analyseur*]	محلّل
Perform [*Réaliser*]	ينجز
Platform [*Plateforme*]	منصة، أرضية
Query [*Requête*]	استعلام
Rendering [*Rendring*]	معالجة
Restrictive [*Restrictive*]	مقتصر، مقيّد
Retrieve [*Extraire*]	استخلاص، استخراج
Scheduled [*Programmé*]	مجدول، مبرمَج
Server [*Serveur*]	خادم، خوادم
Session [*Session*]	جلسة، دورة، حصة
Sheet [*Feuille*]	ورقة
Simulate [*Simuler*]	يحاكي
Static [*Statique*]	ثابت
Style [*Style*]	نمط، طراز
Tags [*Balises, Étiquettes*]	وسوم، علامات
Task [*Tâche*]	مهمة
Template [*Modèle*]	قالب، نموذج
Topic [*Sujet*]	موضوع
Tracking [*Suivi*]	متابعة
Tutorial [*Tutoriel*]	درس توجيهي
Update [*Mise À Jour*]	تحديث
Validity [*Validité*]	صلاحية
Valid [*Valide*]	صالح
View [*Affichage*]	عرض

مرتبة أبجديًا حسب الحرف العربي

Instance [*Instance*]	فرد
Instance [*Instance*]	فرد
Editable [*Éditable*]	قابل للتعديل، للكتابة للتحرير
Template [*Modèle*]	قالب، نموذج
Fragment [*Fragment*]	قطعة، جزء
Object [*Objet*]	كائن، شيء
Dialect [*Dialecte*]	لهجة، لغة فرعية
Bachelor [*Licence*]	ليسانس
Mentoring [*Suivi*]	متابعة، مراقبة
Tracking [*Suivi*]	متابعة
Dynamic [*Dynamique*]	متحرّك، تفاعلي
Inclusive [*Inclusive*]	متضمن
Scheduled [*Programmé*]	مجدول، مبرمَج
Parser [*Analyseur*]	محلّل
Diagram [*Diagramme*]	مخطط
Associated [*Associé*]	مرتبط
Continuous [*Continue*]	مستمر، متواصل
Document [*Document*]	مستند
Rendering [*Rendring*]	معالجة
Average [*Moyenne*]	معدّل
Restrictive [*Restrictive*]	مقتصر، مقيّد
Platform [*Plateforme*]	منصة، أرضية
Task [*Tâche*]	مهمة
Topic [*Sujet*]	موضوع
Feature [*Fonctionnalité*]	ميزة، سمة
Style [*Style*]	نمط، طراز
GUI [*Interface Graphique*]	واجهة المستخدم الرسومية
Interface [*Interface*]	واجهة
Sheet [*Feuille*]	ورقة
Parameters [*Paramètres*]	وسائط، مَعْلمات
Tags [*Balises, Étiquettes*]	وسوم، علامات
Simulate [*Simuler*]	يحاكي
Improve [*Améliorer*]	يحسّن
Customize [*Personnaliser*]	يخصص
Denote [*Noté*]	يرمز
Facilitate [*Faciliter*]	يسهّل
Derive [*Dériver*]	يشتقّ
Ensure [*Assurer*]	يضمن
Handle [*Manipuler*]	يعالج، يتحكم
Enumerate [*Énumérer*]	يعدّ
Assume [*Supposer*]	يفترض
Perform [*Réaliser*]	ينجز

Mandatory [*Obligatoire*]	إجباري
Optional [*Facultatif*]	اختياري
Retrieve [*Extraire*]	استخلاص، استخراج
Query [*Requête*]	استعلام
Assignment [*Affectation*]	إسناد
Notification [*Notification*]	إشعار، تنبيه
Framework [*Framework*]	إطار عمل
Frame [*Cadre*]	إطار
Instead [*Au Lieu*]	بدلا عن
Explicitly [*Explicitement*]	بوضوح
Follow [*Suivant*]	تالي
Update [*Mise À Jour*]	تحديث
Nesting [*Imbrication*]	تداخل، تراكب
Delivery [*Remise*]	تسليم
Design [*Conception*]	تصميم
Overlapping [*Chevauchement*]	تعارض، تداخل
Manipulation [*Manipulation*]	تعامل، معالجة
Edition [*Édition*]	تعديل، تحرير، كتابة
Assessment [*Évaluation*]	تقويم، فحص
Default [*Par Défaut*]	تلقائيا
Execute [*Exécuter*]	تنفيذ، تشغيل
Documentation [*Documentation*]	توثيق
Availability [*Disponibilité*]	توفر
Static [*Statique*]	ثابت
Session [*Session*]	جلسة، دورة، حصة
Assiduity [*Assiduité*]	حضور
Attendance [*Présence*]	حضور
Fields [*Champs*]	حقول
Server [*Serveur*]	خادم، خوادم
Attribute [*Attribut*]	خاصية، سمة
Attribute(class) [*Attribut (classe)*]	خاصية
Method (class) [*Méthode (classe)*]	دالة صنف
Function [*Fonction*]	دالة، وظيفة
Tutorial [*Tutoriel*]	درس توجيهي
Customer [*Client*]	زبون، عميل
Explain [*Expliquer*]	شرح
Valid [*Valide*]	صالح
Validity [*Validité*]	صلاحية
Display [*Affichage*]	عرض
View [*Affichage*]	عرض
Grade [*Grade*]	علامة، درجة
Item [*Élément*]	عنصر
Disadvantages [*Inconvénient*]	عيوب، مساوئ

Bibliography

Abram, David (2026a). *XML Schema Cheat Sheet*. URL: https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/xsd-cheat-sheet.md (visited on 07/23/2026) (cit. on p. 114).

— (2026b). *XML Schema Cheat Sheet (PDF)*. URL: https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/cheat-sheet.pdf (visited on 07/23/2026) (cit. on p. 114).

BaseX Team (2026). *BaseX GUI – High-Performance XML Database Engine*. URL: https://basex.org/basex/gui/ (visited on 07/23/2026) (cit. on p. 114).

Chaudhri, A.B., A. Rashid, and R. Zicari (2003). *XML Data Management: Native XML and XML-enabled Database Systems*. Addison-Wesley. ISBN: 9780201844528. URL: https://books.google.dz/books?id=7LNhdOeQulQC (cit. on p. 116).

Club Scientifique de l'ESI (CSE) (2026). *Trésor ESI: Educational Resources Library*. URL: https://tresor.cse.club/ (visited on 07/23/2026) (cit. on p. 114).

CS Bouira Team (2026). *CS Bouira: Academic Resources for Computer Science Students*. URL: http://csbouira.xyz/ (visited on 07/23/2026) (cit. on p. 114).

Gardarin, Georges (2001). *Bases de données: objet et relationnel*. Paris: Eyrolles. ISBN: 978-2-212-09283-7 (cit. on p. 116).

Michard, A. (2000). *Xml. Langage Et Applications*. Eyrolles. ISBN: 9782212092066. URL: https://books.google.dz/books?id=uMVfAAAACAAJ (cit. on p. 116).

Python Software Foundation (2026). *xml.dom — The Document Object Model API*. URL: https://docs.python.org/3/library/xml.dom.html (visited on 07/23/2026) (cit. on p. 114).

Saxonica (2026). *saxonche: Official Python Package for SaxonC-HE*. Version 13.0.0. URL: https://pypi.org/project/saxonche/ (visited on 07/23/2026) (cit. on p. 114).

Tanzil Project (2026). *Quran Metadata*. URL: https://tanzil.net/docs/quran_metadata (visited on 07/23/2026) (cit. on p. 114).

W3C XML Schema Working Group (2004). *XML Schema Structures Quick Reference*. URL: https://www.cheat-sheets.org/saved-copy/SchemaStructuresQR-2.pdf (visited on 07/23/2026) (cit. on p. 114).

W3C.org (2023). *W3Schools Tutorials*. URL: http://www.w3schools.com (cit. on pp. 45, 113).

W3Schools (2026). *XSLT Elements Reference*. URL: https://www.w3schools.com/xml/xsl_elementref.asp (visited on 07/23/2026) (cit. on p. 113).

XML Copy Editor Project (2026). *XML Copy Editor*. URL: https://xml-copy-editor.sourceforge.io (visited on 07/23/2026) (cit. on p. 113).

Zerrouki, Taha (2026a). *DSSBook Resources: Quran Data*. URL: https://github.com/linuxscout/DSSBook-resources/ (visited on 07/23/2026) (cit. on p. 114).

— (2026b). *Taha Zerrouki — YouTube Channel*. URL: https://www.youtube.com/@taha.zerrouki (visited on 07/23/2026) (cit. on p. 113).

منصة .منصة إلكترونيكس غو Trans. by .بايثون للجميع: التعامل مع البيانات باستخدام لغة بايثون 3 .(2023) تشارلز راسل سيفرانس
إلكترونيكس غو. URL: https://www.py4e.com/translations/AR/ar/book_ar_2023_01_01.pdf (visited on 07/22/2026) (cit. on pp. 45, 113).

دار العربي .90 مشروع بايثون مع الكود المصدري تم حلها وشرحها .بايثون عن طريق الأمثلة .(2022a) د. علاء طعيمة
URL: https://dlarabic.com (cit. on pp. 45, 113).

— (Dec. 2022b). دار العربي .كيف تبرمج في بايثون. URL: https://dlarabic.com (visited on 07/22/2026) (cit. on pp. 45, 113).

جامعة البويرة .نبراس: دليل المصطلحات للشعب التقنية .(2012) طه زروقي (cit. on p. 113).

Trans. .جميل بيلوني Ed. by .تعلم البرمجة وكتابة البرامج وتنقيحها بلغة بايثون .البرمجة بلغة بايثون .(Mar. 20, 2025) ليزا تاغليفيري
by محمد بغات and إيمش .عبد اللطيف .أكاديمية حسوب. URL: https://academy.hsoub.com/files/15-%D8%A7%D9%84%D8%A8%D8%B1%D9%85%D8%AC%D8%A9-%D8%A8%D9%84%D8%BA%D8%A9-%D8%A8%D8%A7%D9%8A%D8%AB%D9%88%D9%86/ (visited on 07/22/2026) (cit. on pp. 45, 113).

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
  <title>...</title>
  <author>...</author>
  <chapter>
    <section>...</section>
  </chapter>
</book>
```

البيانات شبه المهيكلة

تقنيات XML وتطبيقاتها

أصبح التعامل مع البيانات شبه المهيكلة جزءًا أساسيًا في تطوير تطبيقات الويب، وتبادل البيانات بين الأنظمة، وقواعد البيانات، ومعالجة الوثائق الرقمية. ويُعد فهم تقنيات XML من المهارات التي يحتاجها طالب علوم الحاسوب والمطور على حد سواء.

يقدم هذا الكتاب مدخلًا عمليًا ومتدرجًا إلى عالم البيانات شبه المهيكلة، بدءًا من أساسيات XML، مرورًا بـ DTD وXML Schema (XSD)، وXPath وXQuery وXSLT، وصولًا إلى معالجة مستندات XML برمجيًا باستخدام Python.

يعتمد الكتاب على أسلوب التعلم بالممارسة، حيث يضم دروسًا تطبيقية، وأمثلة واقعية، وتمارين، ومشاريع تساعد القارئ على اكتساب المهارات اللازمة لاستخدام تقنيات XML في التطبيقات العملية.

وقد أُعد هذا الكتاب ليلائم المقررات الجامعية، مع المحافظة على طابعه المرجعي الذي يجعله مناسبًا للطلبة، والمهندسين، والمطورين، وكل مهتم بتقنيات البيانات شبه المهيكلة.

```
<data>
  <item>
    <id>1</id>
    <name>...</name>
  </item>
</data>
```