IT3063 – ADVANCED WEB TECHNOLOGIES

XML Schema

Dinoo Gunasekera,
Department of Information Technology,
Faculty of Computing,
General Sir John Kotelawala Defence University

Objectives

- ✓ Understand the purpose of XSD
- ✓ Knowledge about the types of XSDs
- ✓ Understand how to use data types used in XSD
- ✓ Get idea about how to apply restrictions for datatypes

What is XML Schema?

- Defines rules for the structure and content of an XML document.
  - Specify overall structure of an XML document and identify all the components of the XML document
- Documents that adhere to an XML schema are considered valid documents
- W3 recommended

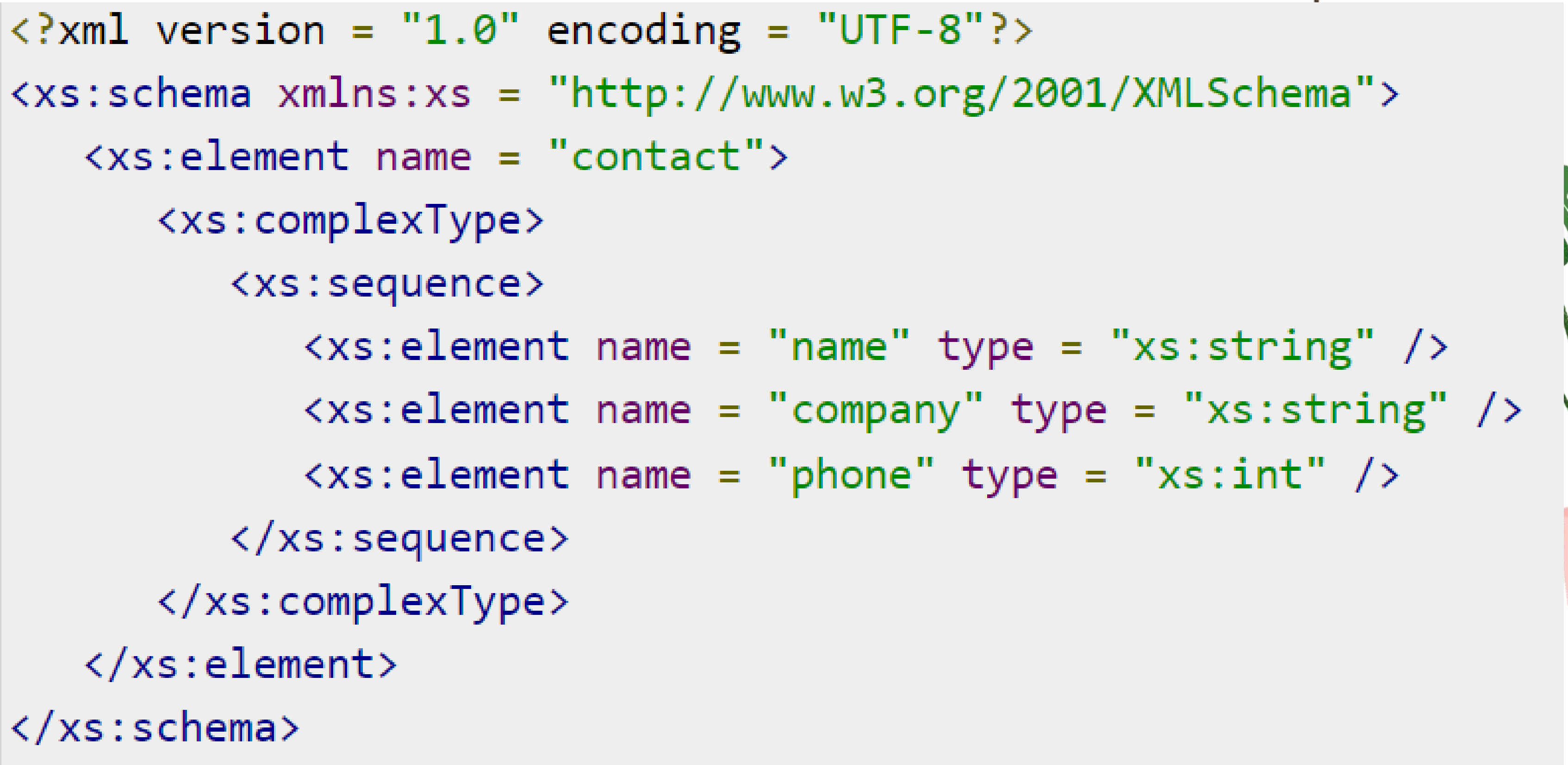
XSD vs DTD

DTD	XSD
Doesn’t support data types	Supports datatypes for elements and attributes
Doesn’t support namespace	Supports namespace
Doesn’t define order for child elements	Defines order for child elements
Not extensible	Extensible
Not simple to learn	Simple to learn
Less control on XML structure	More control on XML structure

XML Schema Lingo

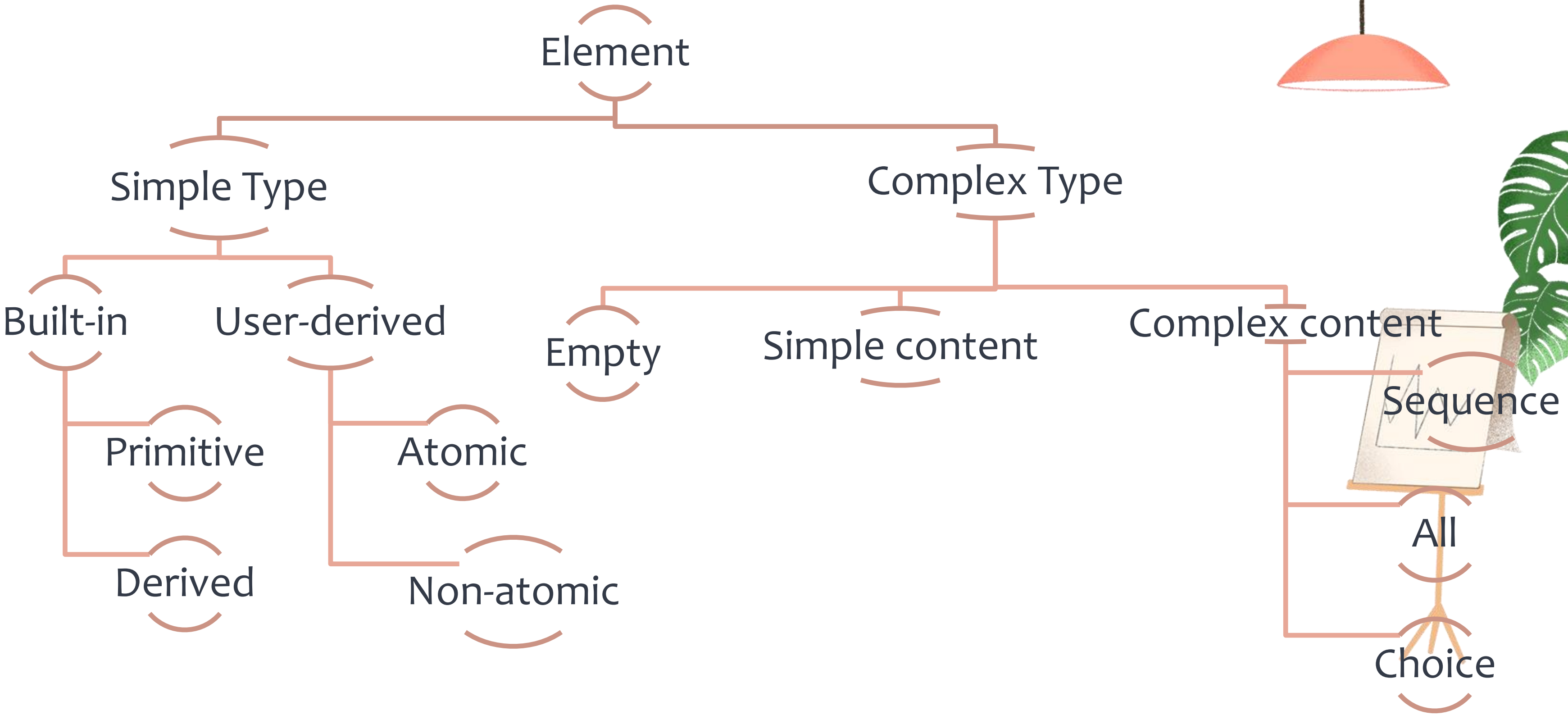
Term	Syntax	Example	What it does
XML declaration	*<xml version="version" encoding="encoding">*	<xml version = "1.0" encoding = "UTF-8">	Tell the processor which version of XML and which character encoding to use
Schema element	<xsd:schema xmlsn:xsd="namespace">	<xsd:schema xmlsn = "http://www.w3.org/2001/XMLSchema">	Identifies the document as XML Schema
Element declaration	<xsd:element name="name">	<xsd:element name="books">	Defines the element named *books*
Attribute declaration	<xsd:attribute name = "name" type = "datatype">	<xsd:attribute name = "sourceType" type="xsd;string">	Defined the attribute named *sourceType*

Syntax Example



```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
   <xs:element name = "contact">
      <xs:complexType>
         <xs:sequence>
            <xs:element name = "name" type = "xs:string" />
            <xs:element name = "company" type = "xs:string" />
            <xs:element name = "phone" type = "xs:int" />
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

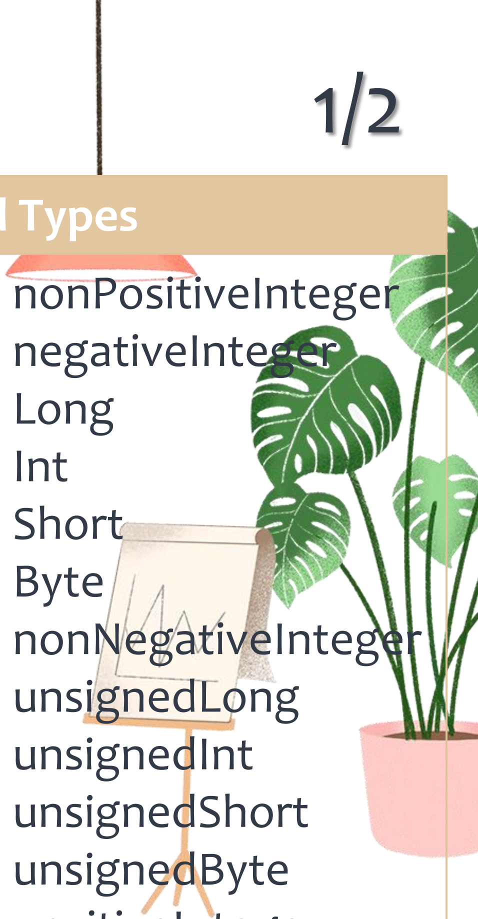
Schema Types



- Element
  - Simple Type
    - Built-in
      - Primitive
      - Derived
    - User-derived
      - Atomic
      - Non-atomic
  - Complex Type
    - Empty
    - Simple content
    - Complex content
      - Sequence
      - All
      - Choice

Built-in Data Types 1/2

Primitive Types		Derived Types	
String	gYearMonth	normalizedString	nonPositiveInteger
Boolean	gYear	Token	negativeInteger
Decimal	gYearMonthDay	Language	Long
Float	gDay	NMTOKEN	Int
Double	gMonth	NMTOKENS	Short
dateTime	hexBinary	Name	Byte
Time	anyURI	NCName	nonNegativeInteger
Date	Qname	ID	unsignedLong
	NOTATION	IDREF	unsignedInt
		IDREFS	unsignedShort
		ENTITY	unsignedByte
		Integer	positiveInteger

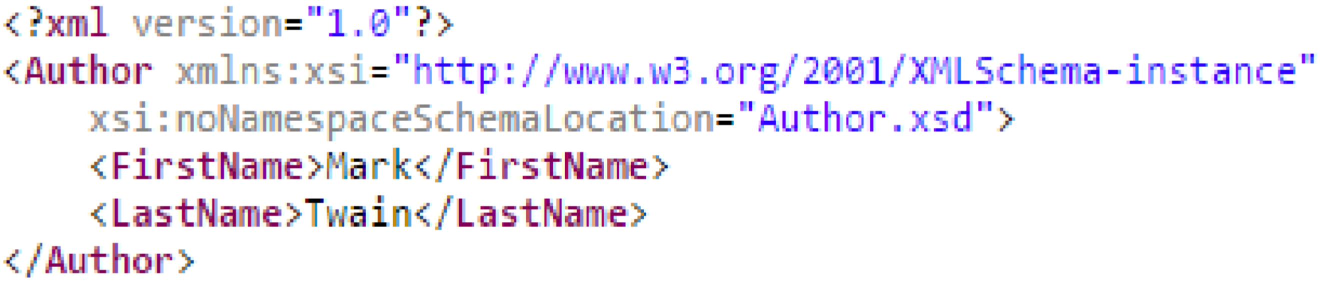


Built-in Data Types 2/2

- string: a collection of characters that is treated as a simple string of text
- decimal: A number that includes a decimal point and some number of decimal places after the point
- dateTime: Can specify what pattern the date and time should use
- anyURI: a URI or URL
- integer: : A number without a decimal point

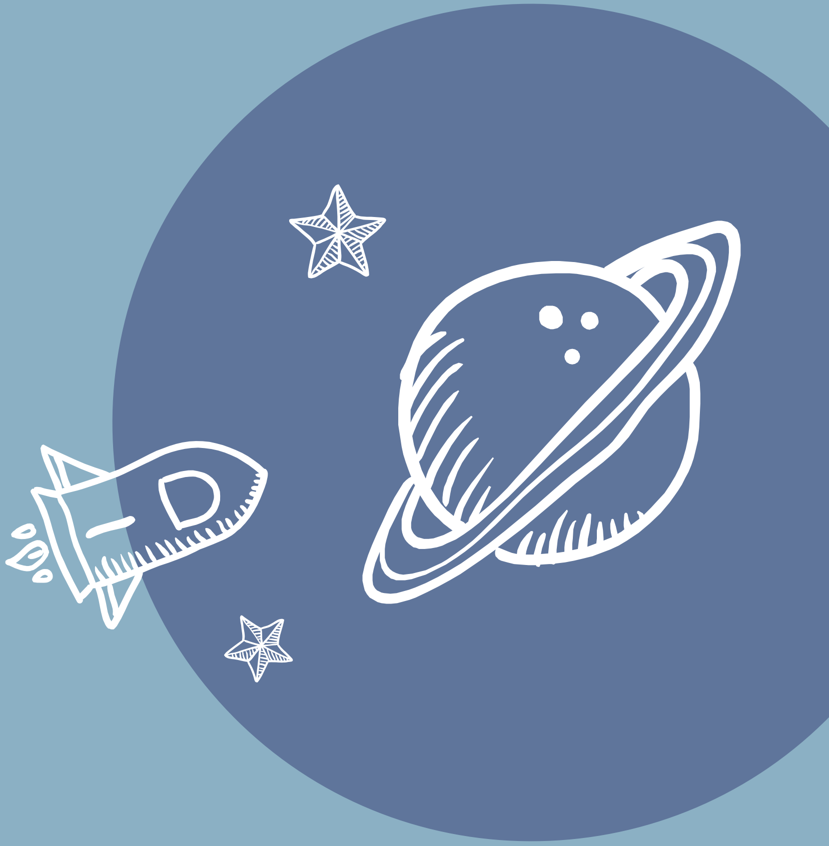
Valid XML Instance

- The **xmlns:xsi** indicated that XML document is an instance of an XML Schema
- **xsi:noNamespaceSchemaLocation** is used to link XML Schema with the XML file.



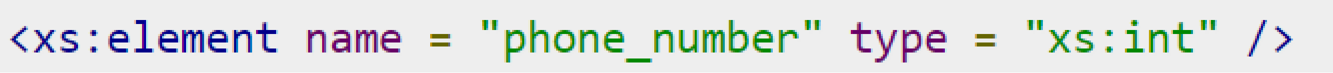
```
<?xml version="1.0"?>
<Author xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="Author.xsd">
    <FirstName>Mark</FirstName>
    <LastName>Twain</LastName>
</Author>
```

Simple Type



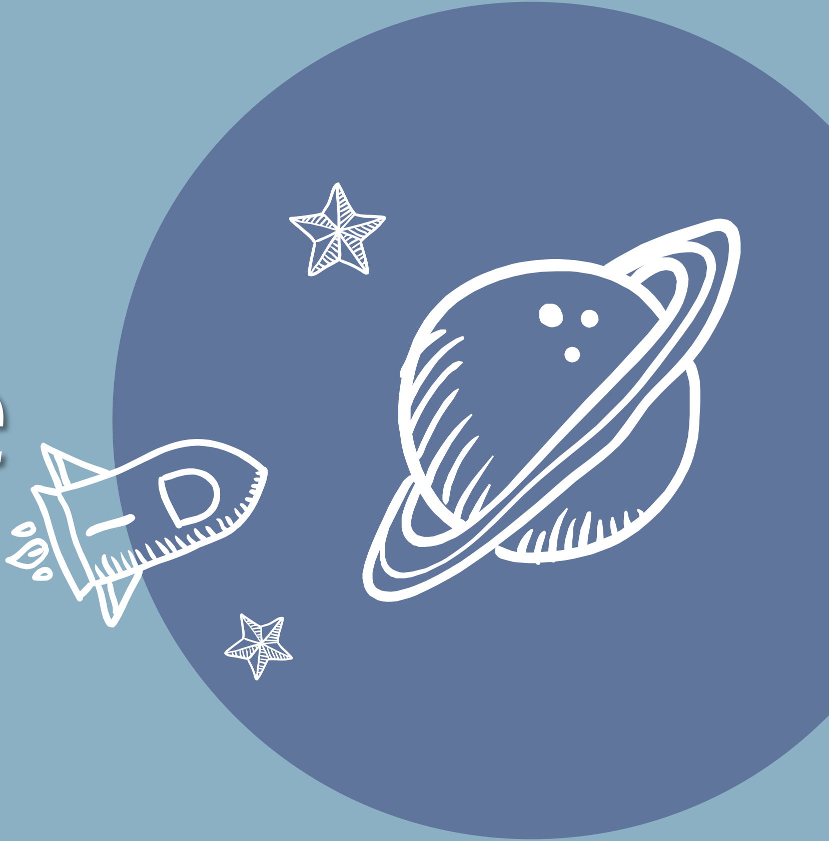
Simple Type

- ✓ Cannot contain any elements nor attributes
- ✓ Used for text-based elements
- ✓ Some of predefined simple types are;
  - ○ xs:integer, xs:boolean, xs:string, xs:date, etc.



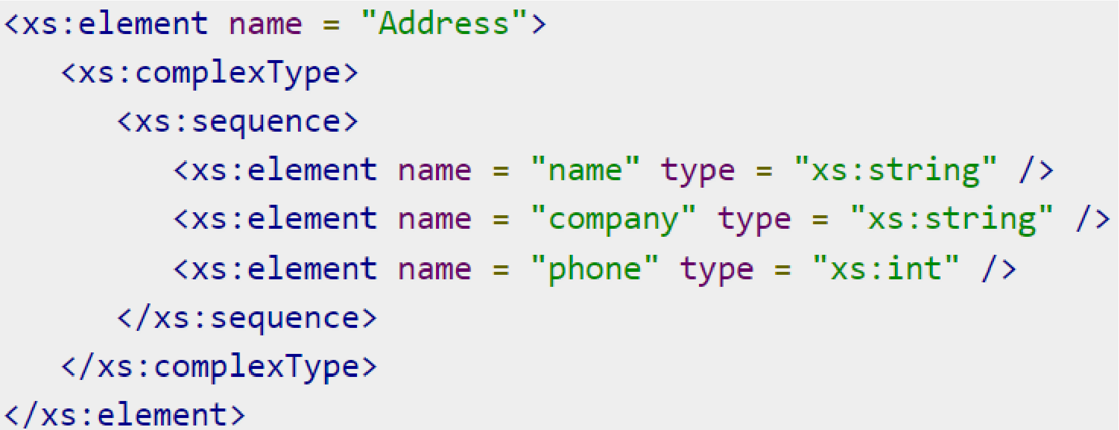
```
<xs:element name = "phone_number" type = "xs:int" />
```

Complex Type



Complex Type

- ✓ Container for other element definitions
- ✓ Allows to specify which child elements an element can contain
- ✓ Provides some structure within XML documents



```
<xs:element name = "Address">
   <xs:complexType>
      <xs:sequence>
         <xs:element name = "name" type = "xs:string" />
         <xs:element name = "company" type = "xs:string" />
         <xs:element name = "phone" type = "xs:int" />
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

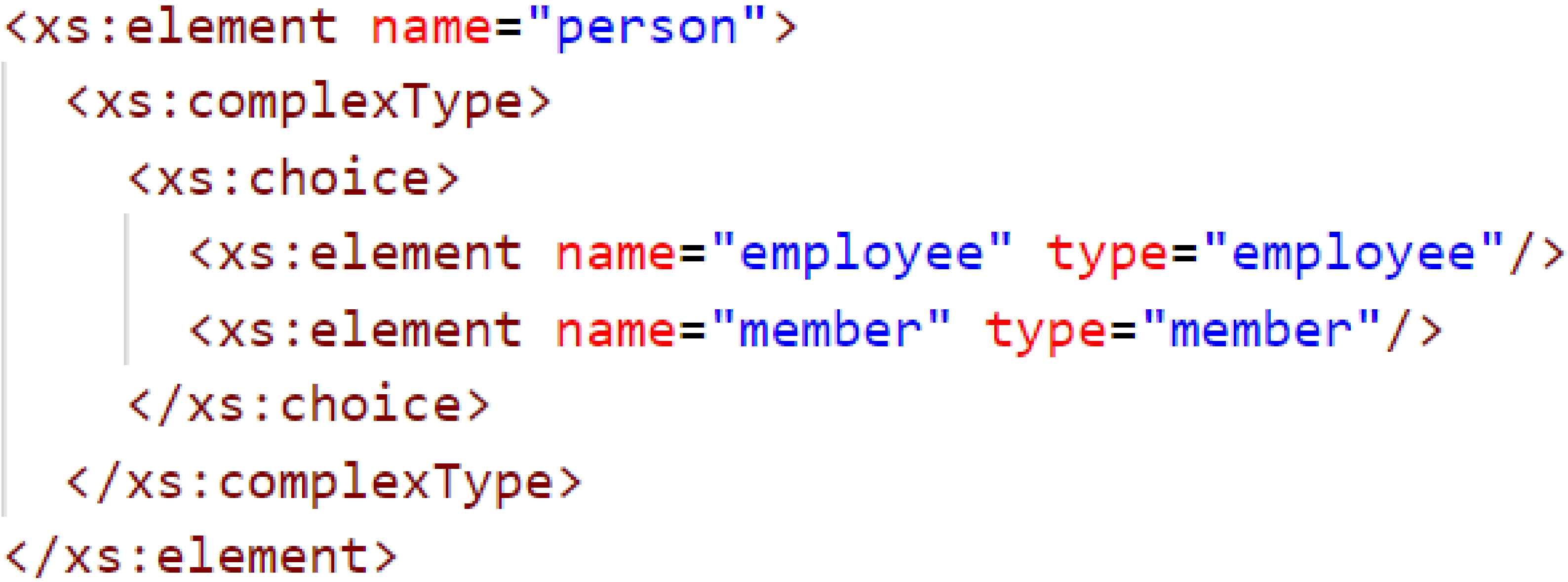
Complex Types

- Empty
- Elements Only
- Text Only
- Mixed
- Indicators
- <any>

Complex Type Indicators

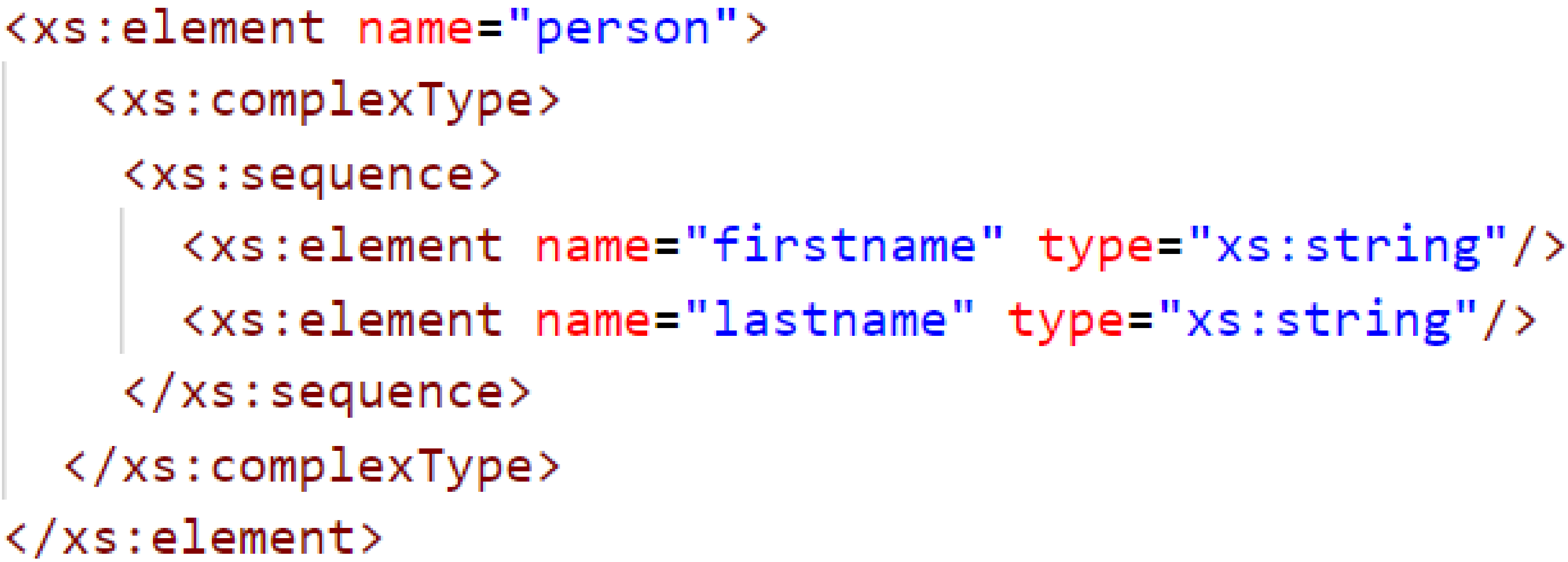
- Choice Indicator
- Sequence Indicator
- All Indicator

Choice Indicator



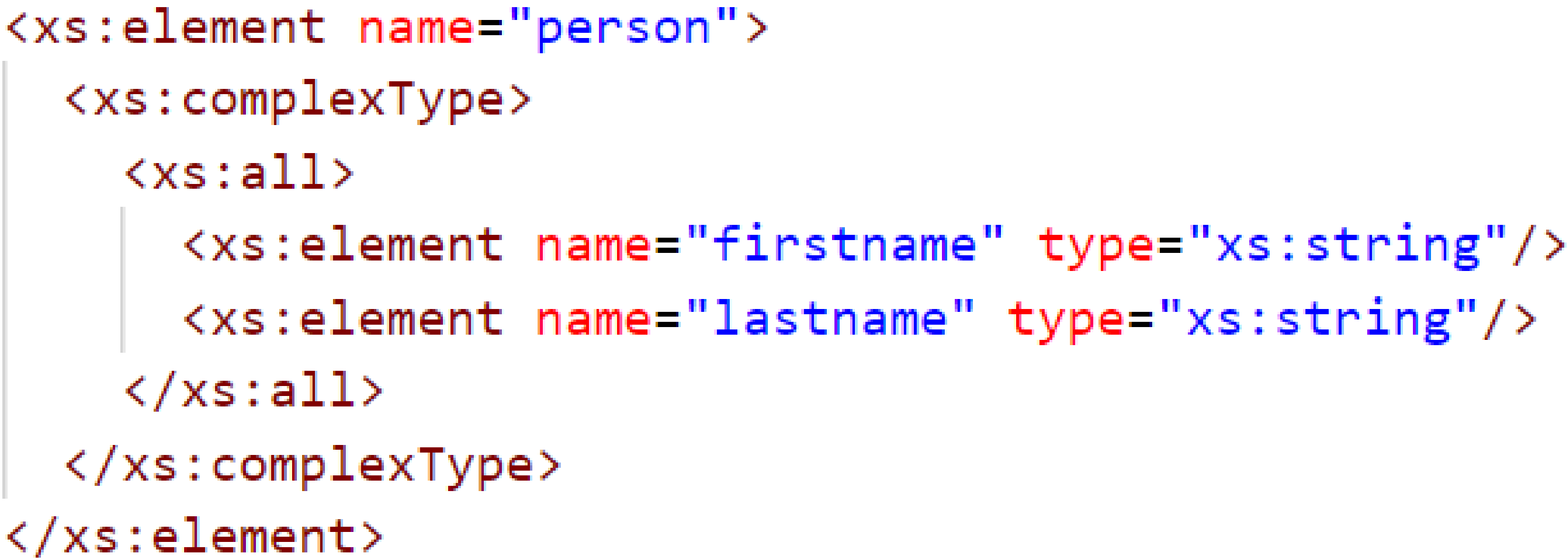
```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="member" type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Sequence Indicator

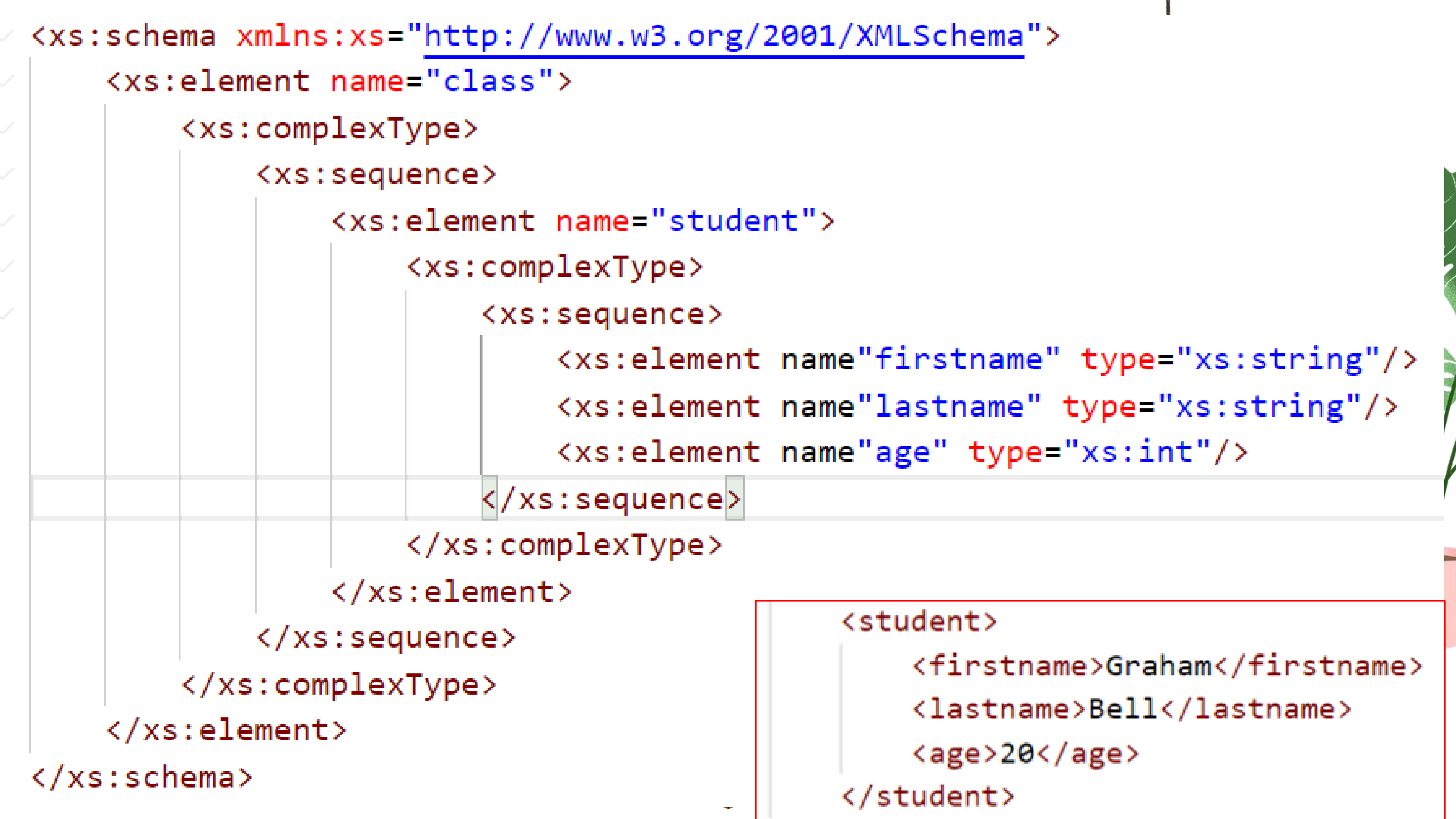


```
<xs:element name="person">
   <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

All Indicator



```
<xs:element name="person">
  <xs:complexType>
    <xs:all>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

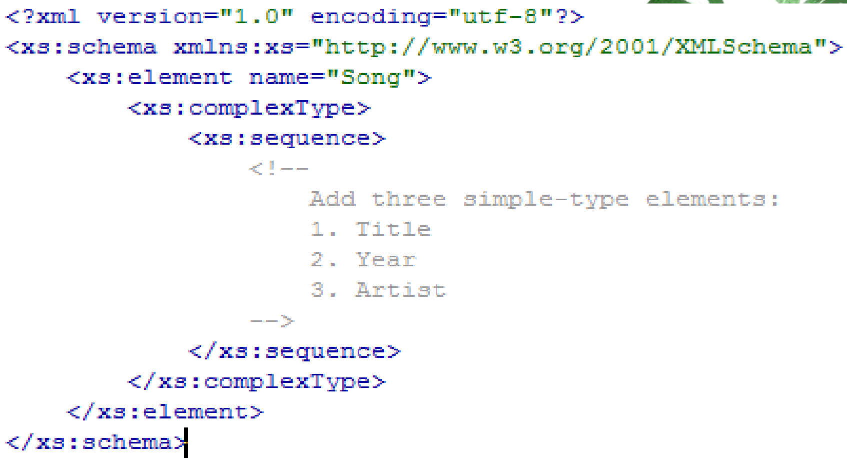


```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="class">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="student">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name"firstname" type="xs:string"/>
                            <xs:element name"lastname" type="xs:string"/>
                            <xs:element name"age" type="xs:int"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

```
    <student>
        <firstname>Graham</firstname>
        <lastname>Bell</lastname>
        <age>20</age>
    </student>
```

Exercise

- ✓ Between the open and close xs:sequence tags, declare three new elements:
  - ○ Title of type xs:string.
  - ○ Year of type xs:gYear.
  - ○ Artist of type xs:string.
- ✓ Save the file.
- ✓ Try to validate the XML against the schema you just created.



```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Song">
        <xs:complexType>
            <xs:sequence>
                <!--
                    Add three simple-type elements:
                    1. Title
                    2. Year
                    3. Artist
                -->
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Exercise

✓ Write a XSD to below XML

```
<?xml version="1.0"?>
  <student>
    <firstname>Graham</firstname>
    <lastname>Bell</lastname>
    <age>20</age>
  </student>
```

Exercise

- Write a XSD to below XML

```
<?xml version="1.0" encoding="utf-8"?>
<home>
    <room>
            <type>livingroom</type>
            <doors>2</doors>
            <windows>3</windows>
            <electricity>
                    <lights>5</lights>
                    <switches>3</switches>
            </electricity>
    </room>
</home>
```

Thank you!

You can find me at:

dgunasekara@kdu.ac.lk

+94711632808