XML and JSON

Topic 5

XML (eXtensible Markup Language)

- It is not a markup language it is a meta-markup language.
- Meta-markup language specifies rules for creating markup languages.
- Provides a simple and universal way of storing and transporting data over the Internet.
- It is both human-readable (self-describing data) and machine-readable.

XML vs HTML

- XML is a meta-markup language
- XML is designed to store and carry data.
- XML tags are not predefined

- HTML is a markup language.
- HTML is designed to display data.
- HTML tags are predefined.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CMPT350>
        <heading> web programming </heading>
        <year> 2020 <semester> winter </semester>
        </year>
        <description>This course focuses on front-end, back-end and
        API technologies for creating web applications.
        </description>
        <instructor> Sara Rouhani</instructor>
</CMPT350>
```

XML Tree Structure

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

XML syntax

- General low-level syntax to expose its rules on all XML documents.
- Document type definitions (DTD)
- XML documents has a single root element
- The document can begin with an <?xml ... ?> header tag ("prolog") (it is an optional field)
- Prolog specifies the version number of the XML standard being used and the encoding.
- XML language has no predefined tag.
- Names are used for naming elements and attributes

XML syntax

- XML elements must have closing tags.
- Names are case sensitive, (Body, BODY and body are distinct)
- Names must begin with a letter or an underscore and can include digits, hyphens, and periods. There is no limitations for names' length.
- Attribute values must be enclosed by single or double quotes.

XML is eXtensible

- An application based on XML works as it is expected even data is added or remove from document.

```
<?xml version="1.0" encoding="UTF-8"?>
<CMPT350>

        <heading> web programming </heading>
        <year> 2020 <semester> winter </semester> </year>
        <instructor> John Doe</instructor>

        <location>

                    <building>Art </building>

                     <room>134</room>

        </location>
</CMPT350>
```

It simplifies platforms changes.

XML advantages:

- It is easy to read and understand.
- It is platform independent
- It is extensible so the XML data stored and transported can be changed without affecting the data presentation.
- Can be used to create new web language

  WSDL for describing available web services

  RSS languages for news feeds

  RDF and OWL for describing resources and ontology

  SMIL for describing multimedia

XML disadvantages

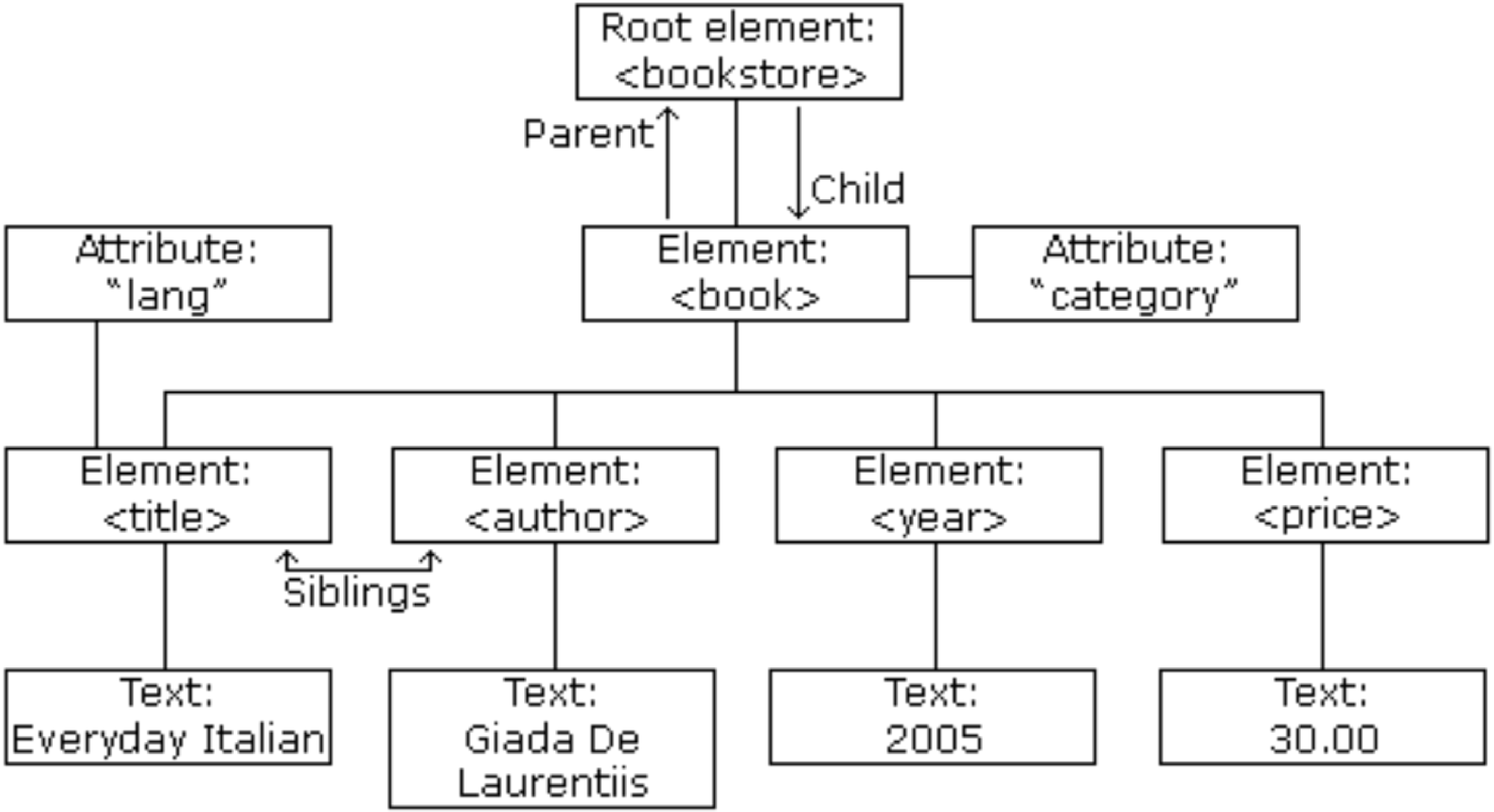
- XML syntax is verbose and bulky and this increases the volume of the data. So, it requires more storage and it increases the transportation cost. It can also effect the performance of the application.

XML separates data from presentation.

- XML does not include any information about displaying the data.
- The same data can be presented in different styles.
- When displaying XML data in HTML, the HTML file should not be edited when the data changes.

-
  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <bookstore>
    <book category="cooking">
      <title lang="en">Everyday Italian</title>
      <author>Giada De Laurentiis</author>
      <year>2005</year>
      <price>30.00</price>
    </book>
    <book category="children">
      <title lang="en">Harry Potter</title>
      <author>J K. Rowling</author>
      <year>2005</year>
      <price>29.99</price>
    </book>
    <book category="web">
      <title lang="en">Learning XML</title>
      <author>Erik T. Ray</author>
      <year>2003</year>
      <price>39.95</price>
    </book>
  </bookstore>
  ```

https://www.w3schools.com/xml/xml_tree.asp



https://www.w3schools.com/xml/xml_tree.asp

XML DOM properties and methods

- Properties:

  firstChild, lastChild, childNodes, nextSibling, previousSibling, parentNode

- Methods:

  appendChild, removeChild, replaceChild

  getElementsByTagName, getAttribute, hasAttributes, hasChildNodes

XML usage in web

- Web publishing
- Web searching
- Metadata
- Databases
- Web services

XML web services

- WSDL

  WSDL is an XML-based language for describing Web services.

- SOAP

  SOAP is an XML based protocol for accessing Web Services

- RDF

  RDF is a framework for describing resources on the web

- RSS

  RSS defines an easy way to share and view headlines and content

WSDL (Web Services Description Language)

```
<definitions>

<types>
  data type used by the web service
</types>

<message>
  definition of the data being communicated
</message>

<portType>
  set of operations that can be performed
</portType>

<binding>
  protocol and data format for each port type
</binding>

</definitions>
```

WSDL example

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

https://www.w3schools.com/xml/xml_wsdl.asp

SOAP (Simple Object Access Protocol)

- It is an application communication protocol.
- It is a format for sending and receiving messages.

SOAP Elements

- Envelope: identifies the XML document as a SOAP message
- Header: Provides header information
- Body: Provides call and response information
- Fault: Presents errors and status information

SOAP

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

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

<soap:Header>
...
</soap:Header>

<soap:Body>
...
  <soap:Fault>
  ...
  </soap:Fault>
</soap:Body>

</soap:Envelope>
```

RDF (Resource Description Framework)

- It is a part of the W3C's **Semantic** Web stack

  Web information has exact meaning

  Web information can be understood and processed by computers

  Computers can integrate information from the web

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

<rdf:RDF
xmlns:rdf="namespace, specifies that elements with the rdf prefix are from the namespace"
></rdf:RDF>

<rdf:Description rdf:about="the url of the web site">
  <si:title>title goes here</si:title>
  <si:author>the name of the author goes here</si:author>
</rdf:Description>
</rdf:RDF>
```

RSS (Really Simple Syndication)

- RSS is used to share content between websites.
- Use Cases: News sites, companies, calendars
- <title> - Defines the title of the channel
- <link> - Defines the hyperlink to the channel
- <description> - Describes the channel

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>Usask CS department homepage</title>
  <link>https://www.cs.usask.ca/</link>
  <description> all information about computer science program at the USASk</description>
  <item>
    <title>Undergraduate program</title>
    <link>https://www.cs.usask.ca/students/undergraduate/undergrad-programs.php</link>
    <description>Undergraduate categories and research focus</description>
  </item>
  <item>
    <title>Graduate program</title>
    <link>https://www.cs.usask.ca/students/graduate/grad-programs.php</link>
    <description>Information about master's and PhD programs</description>
  </item>
</channel>
</rss>
```

JSON

JSON (**J**ava**S**cript **O**bject **N**otation.)

- It is a syntax for storing and exchanging data.
- JSON is text, and any JavaScript object can be converted into JSON.
- JSON is language independent
- A JSON object is a key-value data format that is rendered in curly braces.

JSON Syntax

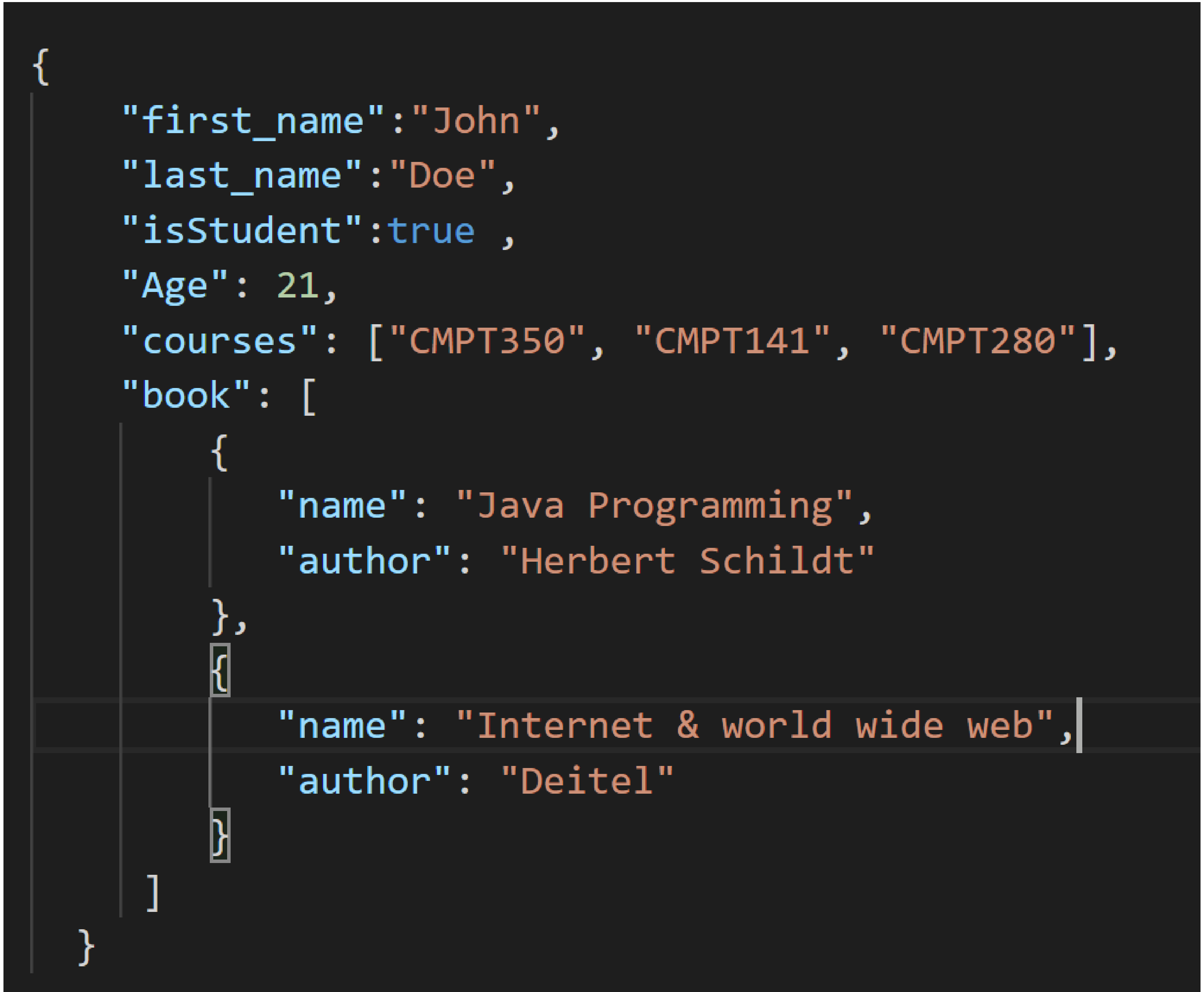
- Data is in name/value pairs
- Data is separated by commas
- The name (key) must be wrapped in double quotes.
- Uses curly braces hold objects
- Square brackets hold arrays

JSON values

- Strings

  must be in double quotes
- Numbers

  must be an integer or a floating point.
- objects
- arrays
- Booleans (true or false)
- null



```
{
    "first_name":"John",
    "last_name":"Doe",
    "isStudent":true ,
    "Age": 21,
    "courses": ["CMPT350", "CMPT141", "CMPT280"],
    "book": [
        {
           "name": "Java Programming",
           "author": "Herbert Schildt"
        },
        {
           "name": "Internet & world wide web",
           "author": "Deitel"
        }
     ]
  }
```

JSON vs XML

Similarities

- self describing
- hierarchical
- can be used by lots of programming languages
- can be fetched with an XMLHttpRequest

Difference

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays
- XML has to be parsed with an XML parser. JSON can be parsed by a standard JavaScript function.

JSON.parse()

- This method converts JSON strings to javascript objects

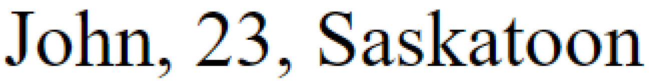
```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var txt = '{"name":"John", "age":23, "city":"Saskatoon"}'
var obj = JSON.parse(txt);
document.getElementById("demo").innerHTML = obj.name + ",
" + obj.age + ", " + obj.city;

</script>

</body>
</html>
```



John, 23, Saskatoon

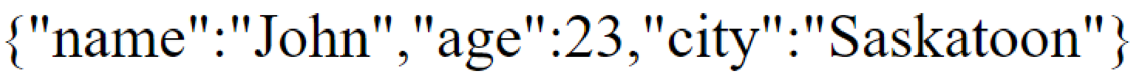
JSON.stringify()

- This method converts a JavaScript object into a string.

```
<!DOCTYPE html>
<html>
<body>
<p id="p1"></p>

<script>
var obj = { name: "John", age: 23, city: "Saskatoon" };
var myJSON = JSON.stringify(obj);
document.getElementById("p1").innerHTML = myJSON;
</script>

</body>
</html>
```



```
{"name":"John","age":23,"city":"Saskatoon"}
```