Workshop 3

COMP20008 Elements of Data Processing

Learning outcomes

By the end of this class, you should be able to:

- write valid XML and JSON documents
- explain differences between XML and JSON
- read and write XML and JSON documents in Python
- recognise and classify types of data errors

XML partial solutions

Q1: HTML vs XML

- Both are web markup languages
- Both originate from SGML (Standardized General Markup Language)
- Both use tags as basic building blocks

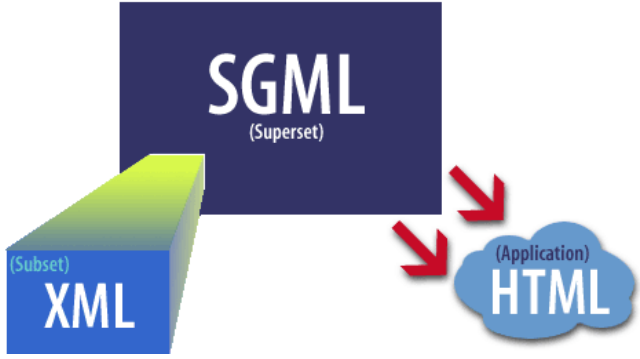


Image source: https://www.j2eeonline.com/xml-programming/module2/relationship-between-metalanguages.jsp

Q1: HTML vs XML

	XML	HTML
Purpose	Transferring information	Presenting information
User-defined tags	Yes	No
Case sensitive tags	Yes	No
White-space	Can be preserved	Not preserved
Closing tags	Compulsory	Optional

Q2: XML syntax errors

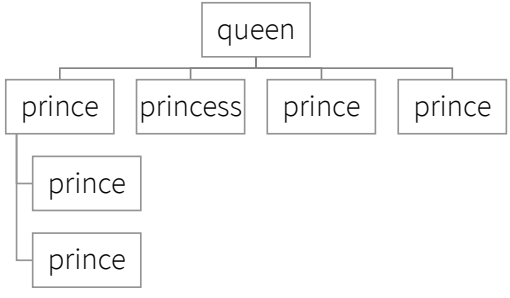
```
<?xml version="1.0" encoding="utf-8"?>
<queen title="Queen Elizabeth II" marriedTo="Philip, Duke of Edinburgh">
  <prince title="Charles, Prince of Wales" marriedTo="Lady Diana Spencer">
    <prince title="Prince William of Wales" />
    <prince title="Prince Henry of Wales" />
  </Prince>
  <princess title="Anne, Princess Royal" />
  <prince title="Andrew, Duke of York" />
  <prince title="Edward, Earl of Wessex" >
</queen>
```

Q2: XML syntax errors

```
<?xml version="1.0" encoding="utf-8"?>
<queen title="Queen Elizabeth II" marriedTo="Philip, Duke of Edinburgh">
  <prince title="Charles, Prince of Wales" marriedTo="Lady Diana Spencer">
    <prince title="Prince William of Wales" />
    <prince title="Prince Henry of Wales" />
  </prince>
  <princess title="Anne, Princess Royal" />
  <prince title="Andrew, Duke of York" />
  <prince title="Edward, Earl of Wessex" />
</queen>
```

Q3: XML elements and attributes

```
<?xml version="1.0" encoding="utf-8"?>
<queen title="Queen Elizabeth II"
marriedTo="Philip, Duke of Edinburgh">
  <prince title="Charles, Prince of Wales"
marriedTo="Lady Diana Spencer">
    <prince title="Prince William of Wales" />
    <prince title="Prince Henry of Wales" />
  </prince>
  <princess title="Anne, Princess Royal" />
  <prince title="Andrew, Duke of York" />
  <prince title="Edward, Earl of Wessex" />
</queen>
```



Q3: XML elements and attributes

```xml
<person>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
  <gender>female</gender>
</person>
```

Gender as *element*

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

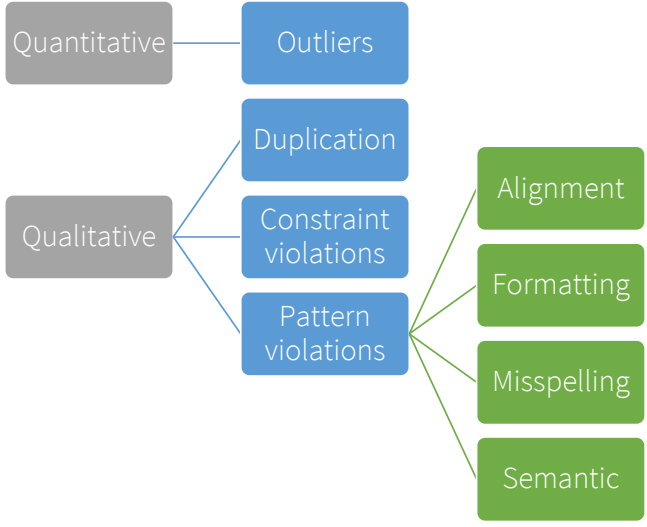
Gender as *attribute*

- Elements can contain
  - Text
  - Attributes
  - Other elements
  - Or a mix of the above
- Attributes are less flexible
  - Text only + not extensible

Preprocessing partial solutions

Q7: Data errors

- Range errors
  - 201 m for human height
  - -1 for month
- Semantic errors
  - Mixing temperatures in Celsius and Fahrenheit
  - “Unimelb” vs “University of Melbourne” vs “Melbourne University”
- Format errors
  - Street, ST, Str
  - +61431123456, 0431123456



Source: Abedjan, Ziawasch, et al. "Detecting data errors: Where are we and what needs to be done?." *Proceedings of the VLDB Endowment* 9.12 (2016): 993-1004.

Q7: Data errors

In US smoking data (1995-2010):

- Range errors
  - Cell A5: year in the future (guess correct value is 2010 based on context)
  - Cell E8: percentage over 100 (impute by ensuring row sums to 100%)
- Semantic errors
  - Cells B2, B3, B4: abbreviation instead of state name (can fix)
- Format errors
  - Cell E16: text instead of number (can fix)
- Duplication
  - Rows 12 and 512: duplicates (suspect A12 should be 2010)

Q8: Data cleaning tools/methods

- A custom Python script
  - Use logical selection to detect errors
  - `Series.replace()` replace known variations by looking up canonical values in a dictionary
  - `Series.value_counts()` to find semantic and format errors
  - `Series.to_numeric()` will raise an error if a string can't be cast to a number
  - `DataFrame.duplicated()` will reveal duplicate rows
- Data cleaning tools like OpenRefine

Additional slides

XML quiz

Source: https://www.w3schools.com/quiztest/quiztest.asp?qtest=XML

XML is a replacement for HTML.

a) True
b) False

XML is a replacement for HTML.

a) True
b) False

What's the correct syntax of the XML Prolog?

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

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

c) `<xml version="1.0" encoding="UTF-8" />`

What's the correct syntax of the XML Prolog?

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

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

c) `<xml version="1.0" encoding="UTF-8" />`

Is the XML document below well-formed?

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

a) Yes

b) No

Is the XML document below well-formed?

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

a) Yes

b) No—missing root element

Which statement is NOT true?

a) XML documents must have a root tag
b) White-space is not preserved in XML
c) XML tags are case sensitive
d) XML elements must be properly nested

Which statement is NOT true?

a) XML documents must have a root tag
b) White-space is not preserved in XML
c) XML tags are case sensitive
d) XML elements must be properly nested

Which of the following are invalid names for an XML element?

a) `<1dollar>`

b) `<h1>`

c) `<Note>`

d) `<first name>`

e) `<NAME>`

Which of the following are invalid names for an XML element?

a) `<1dollar>`

b) `<h1>`

c) `<Note>`

d) `<first name>`

e) `<NAME>`

JSON quiz

Which of the following statements are incorrect?

a) Both JSON and XML are hierarchical
b) Both JSON and XML use end tags
c) Both JSON and XML are interoperable (language-independent)
d) Both JSON and XML are human-readable
e) Both JSON and XML natively support arrays

Which of the following statements are incorrect?

a) Both JSON and XML are hierarchical
b) Both JSON and XML use end tags
c) Both JSON and XML are interoperable (language-independent)
d) Both JSON and XML are human-readable
e) Both JSON and XML natively support arrays

Which of the following is NOT a valid JSON object?

a)

```
{ "name": “Tweety",
  "age": 76,
  "phone": "888-123-4567",
  "email": “tweety@xyz.com",
  "happy": true }
```

b)

```
{ "name": “Tweety",
  "age": 76,
  "phone": "null",
  "email": "null",
  "happy": "true" }
```

c)

```
{ "name": “Tweety",
  "age": 76,
  "phone": {"888-123-4567", "888-765-4321"},
  "email": "null",
  "happy": "true" }
```

Which of the following is NOT a valid JSON object?

a)

```
{ "name": “Tweety",
  "age": 76,
  "phone": "888-123-4567",
  "email": “tweety@xyz.com",
  "happy": true }
```

b)

```
{ "name": “Tweety",
  "age": 76,
  "phone": "null",
  "email": "null",
  "happy": "true" }
```

c)

```
{ "name": “Tweety",
  "age": 76,
  "phone": {"888-123-4567", "888-765-4321"},
  "email": "null",
  "happy": "true" }
```

Which of the following is NOT a valid JSON value?

a) a string
b) a number
c) an object (JSON object)
d) an array
e) a boolean
f) null
g) none of the above

Which of the following is NOT a valid JSON value?

a) a string
b) a number
c) an object (JSON object)
d) an array
e) a boolean
f) null
g) none of the above