

# INFO-H-509 : Technologies XML

## TP 1 - Corrigé

Professeur : Stijn Vansummeren

Teaching Assistant : Michaël Waumans

<http://cs.ulb.ac.be/public/teaching/infoh509>

---

### XML Well-formedness

```
<?xml version="1.0" encoding="utf-8" ?>
<catalogue>
  <course mnemonic="INFO-H-509">
    <title>Technologies XML</title>
    <professor email="stijn.vansummeren@ulb.ac.be">Stijn Vansummeren</professor>
    <assistant email="fpicalau@ulb.ac.be">François Picalausa</assistant>
    <weight>3 <unit type="ECTS" /></weight>
  </course>
  <course mnemonic="INFO-H-302">
    <!-- Ce cours ne sera plus donné l'année prochaine -->
    <title>Analyse & conception par object</title>
    <professor email="ezimanyi@ulb.ac.be">Esteban Zimányi</professor>
    <assistant email="fservais@ulb.ac.be">Frédéric Servais</assistant>
    <assistant email="sboucher@ulb.ac.be">Serge Boucher</assistant>
    <weight>3 <unit type="ECTS" /></weight>
  </course>
  <course mnemonic="INFO-H-303">
    <title>Bases de données</title>
    <professor email="ezimanyi@ulb.ac.be">Esteban Zimányi</professor>
    <assistant email="fservais@ulb.ac.be">Frédéric Servais</assistant>
    <weight>6 <unit type="ECTS"/></weight>
  </course>
  <person email="stijn.vansummeren@ulb.ac.be" laboratory="WIT">
    <name>Stijn Vansummeren</name>
    <interest>Systèmes d'information</interest>
    <interest>Bases de données scientifiques</interest>
    <interest>Web sémantique</interest>
  </person>
  <person email="ezimanyi@ulb.ac.be" laboratory="WIT">
    <name>Esteban Zimányi</name>
    <interest>Systèmes d'information</interest>
    <interest>Entrepôts de données</interest>
    <interest>Web sémantique</interest>
  </person>
  <person email="fpicalau@ulb.ac.be" laboratory="WIT">
    <name>François</name>
    <interest>Web sémantique</interest>
  </person>
```

</catalogue>

Notice that the attributes cannot be multivalued. Also, an attribute cannot be decomposed into multiple fields, like an adress. Except for these constraints, the choice between elements or attributes is mostly arbitrary.

## Unicode

### Exercise 1.4

Give the character associated to the *code point* U+265C (name and glyph).

It is a character representing the black tower in chess (BLACK CHESS ROOK).

**Additional exercise :** Give the UTF-8 encoding of this character.

The code point indicates that the character is encoded by 3 code units. Also  $265C_{16} = 10\ 0110\ 0101\ 1100_2$ . This gives us the following encoding 11100010 10011001 10011100.

### Exercise 1.5

Give the code points encoded by the UTF-8 sequence 00110100 00110010 11100010 10000000 10100110.

When the first bit of a code unit is 0, this emans that it encodes only one character. If it starts with 1110, it means that it is part of a group of three consecutive code units which encode only one character. We have then :

- Code unit 1 :  $011\ 0100_2 = 34_{16}$  is the character “4”,
- Code unit 2 :  $011\ 0010_2 = 32_{16}$  is the character “2”.
- Code units 3 to 5 :  $0010\ 000000\ 100110_2 = 2026_{16}$  refers to the character HORIZONTAL ELLIPSIS “...”.

The complete text is “42...”

## XPath requests

### Exercise 1.6

1. `//Customer[@CustomerID="HUNGC"]//Country`  
Returns the country in which the client "HUNGC" is based.
2. `/Root/Orders//ShipCountry | /Root/Customers//Country`  
The list of countries of the clients as well as the delivery countries.
3. `//ShipCity[following-sibling::ShipRegion eq "OR"]`  
For all the deliveries in Oregon, returns the city of the delivery.
4. `//ShipCity[following-sibling::ShipRegion is "OR"]`  
Error : the operator is compares nodes. "OR" is a litteral (of the type xs:string) and not a node.
5. `/Root/Customers/Customer/Phone[1]`  
Returns the first phone number of each client.
6. `(/Root/Customers/Customer/Phone)[1]`  
Rerturns the first phone number in the document.

## Exercise 1.7

It is always more efficient to use the complete path instead of the axe `descendant-or-self` or its abbreviation `//`, despite the commodity of the last. To improve the readability, we will, in the following, neglect this rule of good practise.

1. Customers who have the title Marketing Manager.

```
/Root/Customers/Customer[ContactTitle eq "Marketing Manager"]
```

**Remark :** We could use `=` instead of `eq`. However, as we know that the element `ContactTitle` is unique for a client, using `eq` eventually allows the XPath engine to optimize the request.

2. The HTML elements of this document.

```
//*[namespace-uri() eq "http://www.w3.org/1999/xhtml"]
```

**Remark :** We could also use `//html:*`, under one condition. That we devine a corresponding namespace in the XPath engine. Indeed, the `html` prefix defined in the document is totally independant of the prefixes known to the Xpath engine. This allows the XPath expressions to stay valid even in the case of prefix modifications in the document.

3. Customers whose name *contains* Yoshi.

```
/Root/Customers/Customer[contains(CustomerName, "Yoshi")]
```

4. The highest, lowest and average freight costs. (Element Freight)

```
(max(//Freight), min(//Freight), avg(//Freight))
```

5. Orders that have not yet been sent (attribute shippedDate is missing for shipInfo)

```
//ShipInfo[not(@ShippedDate)]
```

6. The first order (in document order), for each customer.

```
//Order[not(preceding-sibling::Order/CustomerID = CustomerID)]
```

7. Orders made by Greal in April 1998.

```
//Order[CustomerID="GREAL"  
  [xs:dateTime("1998-04-01T00:00:00") le xs:dateTime(OrderDate)]  
  [xs:dateTime(OrderDate) lt xs:dateTime("1998-05-01T00:00:00")]
```

8. Orders made by a customer based in California.

```
//Order[CustomerID =  
  /Root//Customer[FullAddress/Region eq "CA"]/@CustomerID]
```

9. The countries of delivery (or states for America).

```
for $country in //ShipCountry return  
  if ($country eq "USA") then  
    $country/preceding-sibling::ShipRegion/text()  
  else  
    $country/text()
```

Or even :

```
//ShipInfo[ShipCountry eq "USA"]/ShipRegion/text() |  
//ShipInfo[ShipCountry ne "USA"]/ShipCountry/text()
```

10. The query that tests whether there is a command supplied by delivery man 3.

```
some $x in //ShipVia satisfies $x=3
```