

Started on	Monday, 1 July 2024, 2:38 PM
State	Finished
Completed on	Monday, 1 July 2024, 2:45 PM
Time taken	7 mins 30 secs
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 5.00 out of 5.00

Write a Python code the build the following XML structure. Print/output it with the dump() function.

```
<tag1>
  <tag2>Animal</tag2>
  <tag3>Domestic</tag3>
  <tag4>
    <tag4.1>
      Cat
    </tag4.1>
    <tag4.2>
      Persian
    </tag4.2>
  </tag4>
  <tag5>Iran</tag5>
  <tag6>Male</tag6>
  <tag7>2021.05.04</tag7>
</tag1>
```

Note: In Python ElementTree library, we can add text to an XML element/subelement as follows.  
Assuming ele is an element or subelement,  
ele.text = "MyText"

Above line adds "MyText" as the text of element ele.

Answer: (penalty regime: 0 %)

Reset answer

```
1 import xml.etree.ElementTree as ET
2
3 # Create the root element
4 tag1 = ET.Element("tag1")
5
6 # Create sub-elements and assign text values
7 tag2 = ET.SubElement(tag1, "tag2")
8 tag2.text = "Animal"
9
10 tag3 = ET.SubElement(tag1, "tag3")
11 tag3.text = "Domestic"
12
13 tag4 = ET.SubElement(tag1, "tag4")
14
15 tag4_1 = ET.SubElement(tag4, "tag4.1")
16 tag4_1.text = "Cat"
17
18 tag4_2 = ET.SubElement(tag4, "tag4.2")
19 tag4_2.text = "Persian"
20
21 tag5 = ET.SubElement(tag1, "tag5")
22 tag5.text = "Iran"
23
24 tag6 = ET.SubElement(tag1, "tag6")
```

	Expected	Got	
✓	<tag1><tag2>Animal</tag2><tag3>Domestic</tag3> <tag4><tag4.1>Cat</tag4.1><tag4.2>Persian</tag4.2> </tag4><tag5>Iran</tag5><tag6>Male</tag6> <tag7>2021.05.04</tag7></tag1>	<tag1><tag2>Animal</tag2><tag3>Domestic</tag3> <tag4><tag4.1>Cat</tag4.1><tag4.2>Persian</tag4.2> </tag4><tag5>Iran</tag5><tag6>Male</tag6> <tag7>2021.05.04</tag7></tag1>	✓

Passed all tests! ✓

Correct

Marks for this submission: 5.00/5.00.

Question 2

Correct

Mark 5.00 out of 5.00

Shown below is an XML document showing vehicle information.

```
<motorvehicles>
  <vehicle>
    <registration_no>CBB1456</registration_no>
    <make>Toyota</make>
    <model>Premio</model>
  </vehicle>
  <vehicle>
    <registration_no>PR2245</registration_no>
    <make>Mazda</make>
    <model>Bongo</model>
  </vehicle>
  <vehicle>
    <registration_no>DE2115</registration_no>
    <make>TATA</make>
    <model>Sumo</model>
  </vehicle>
  <vehicle>
    <registration_no>CAR7785</registration_no>
    <make>Kia</make>
    <model>Optima</model>
  </vehicle>
</motorvehicles>
```

Write a python program to update the details of the vehicle with the registration number DE2115 as follows:

```
<registration_no>DE2115</registration_no>

<make>Nissan</make>

<model>Skyline</model>
```

For your convenience, the above XML document is preloaded as a string to the coding window.

After updating the XML, your program should print the Registration Numbers of each Nissan vehicle. (Registration Numbers should be printed line by line.)

Answer: (penalty regime: 0 %)

Reset answer

```
11 <registration_no>PR2245</registration_no>
12 <make>Mazda</make>
13 <model>Bongo</model>
14 </vehicle>
15 <vehicle>
16 <registration_no>DE2115</registration_no>
17 <make>TATA</make>
18 <model>Sumo</model>
19 </vehicle>
20 <vehicle>
21 <registration_no>CAR7785</registration_no>
22 <make>Kia</make>
23 <model>Optima</model>
24 </vehicle>
25 </motorvehicles>'''
26
27 # Parse the XML data
28 root = ET.fromstring(xml_data)
29
30 # Update the details of the vehicle with registration number DE2115
31 for vehicle in root.findall('vehicle'):
32     reg_no = vehicle.find('registration_no').text
33     if reg_no == 'DE2115':
34         vehicle.find('make').text = 'Nissan'
```

	Expected	Got	
✓	DE2115	DE2115	✓

Passed all tests! ✓

Correct

Marks for this submission: 5.00/5.00.

◀ 4.5 Introduction to XML with Python

Jump to...

4.6 Introduction to JSON with Python ▶


GET IN TOUCH

🏠 University of Moratuwa  
Centre for Open & Distance Learning  
CODL

☎ 011 308 2787/8  
☎ 011 265 0301 ext. 3850,3851  
✉ open@uom.lk

🌐 [University Website](#)  
🌐 [CODL Website](#)

[Data retention summary](#)

 mora\_logo.png Sponsored By  logo.png

[f\\_](#) [t\\_](#) [g\\_](#) [i\\_](#) [in\\_](#) [globe](#)