

Web Languages and Technologies

Faculdade de Engenharia da Universidade do Porto
7th January 2013

Duration: 2h

Consultation of annotations in paper format allowed

1. Given the HTML and CSS snippets presented below:

```
1 <body>
2   <div id="a1" class="a2 x">A</div>
3   <div id="b1" class="b2 x">B</div>
4   <div id="c1" class="c2 y">C</div>
5   <div id="d1" class="d2 y">D</div>
6 </body>
```

Listing 1: HTML Snippet

```
1 div {
2   border: 1px solid black;
3   width: 100px;
4   height: 100px;
5   text-align: center;
6   background-color: cyan;
7 }
8
9 #a1 { float: left; }
10 #b1 { float: left; }
11 #c1 { float: left; clear: left;}
12 #d1 { float: left;}
13
14 body>:first-child, .b2+div { background-color: yellow; } /* R1 */
15 div#a1, .c2 { background-color: green; } /* R2 */
16 #a1, #c1.c2 { background-color: blue; } /* R2 */
17 body>#d1, .x+div { background-color: red; } /* R4 */
```

Listing 2: CSS Snippet

1 val.

- (a) Calculate the specificity of each of the CSS selectors in the rules marked as R1 through R4.

2½ val.

- (b) Do an approximate drawing of what would be presented in a browser. Draw **5 diagrams**: one without taking into account any of the 4 rules marked as R1 through R4; then draw another four diagrams, having each one of the rules from R1 to R4 being added in each one of them (*i.e.* in the second diagram only rule R1 should be taken into account, in the third only rules R1 and R2 and in the fourth rules R1, R2 and R3, ...). Write in each *div* the corresponding background color.

2. Write a piece of code in *jQuery* that, when applied to the code of the previous question, will yield the following result:

2 val.

- (a) By clicking in any of the *divs*, an Ajax request should be made to the URL `http://getcolor.com/` containing a variable, passed by GET, with the name *id* and value equal to the *id* of the clicked *div*. When receiving a response, the background color of the *div* should be changed to the received color. Consider that the response has the following format in JSON:

```
1 {
2   "id": "a1",
3   "color": "#336699"
4 }
```

1½ val.

(b) While the mouse pointer is over any of the *divs*, the document should have a background color equal to the color of the next *div*. If the mouse lies on the last *div*, then the color of the first one should be used.

3. Considerer a string containing “**apples bananas pears bananas oranges**”. Write down the first substring that matches with each one of the following regular expressions:

1 val.

(a) `/(\w{5}\s){2}/`

1 val.

(b) `/n(?:a)/`

1 val.

(c) `/(\w{3,}).*\1/`

1 val.

(d) `/(\w{3,}).*(?=\1)/`

Note: the character `/` is not part of the regular expression, it is just the delimiting character.

4. Consider the following XSD document that is stored in a file called *school.xsd*:

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <xs:schema
3   targetNamespace="http://exame.ltw/2013"
4   xmlns:xs="http://www.w3.org/2001/XMLSchema"
5   xmlns="http://exame.ltw/2013"
6   elementFormDefault="qualified"
7   attributeFormDefault="unqualified">
8
9 <xs:element name="school" type="school" />
10
11 <xs:complexType name="school">
12   <xs:sequence>
13     <xs:element name="students" maxOccurs="1">
14       <xs:complexType>
15         <xs:sequence maxOccurs="3">
16           <xs:element name="student" type="student" />
17         </xs:sequence>
18       </xs:complexType>
19     </xs:element>
20   </xs:sequence>
21 </xs:complexType>
22
23 <xs:complexType name="student">
24   <xs:sequence>
25     <xs:element name="grade" type="grade" maxOccurs="2" />
26   </xs:sequence>
27   <xs:attribute name="name" type="xs:string" use="required" />
28   <xs:attribute name="age" type="xs:integer" />
29 </xs:complexType>
30
31 <xs:complexType name="grade">
32   <xs:attribute name="class" type="xs:string" use="required" />
33   <xs:attribute name="value" type="value" use="required" form="qualified" />
34   <xs:attribute name="state" fixed="valid" />
35 </xs:complexType>
36
37 <xs:simpleType name="value">
38   <xs:restriction base="xs:string">
39     <xs:pattern value="[A-F]" />
40   </xs:restriction>
41 </xs:simpleType>
42 </xs:schema>
```

Listing 3: Documento XSD

2 val.

(a) Write a valid XML document, that conforms to the XSD, that uses all the possibilities described in it (*i.e.* if an element can appear a maximum of three times, the document should contain

that element three times; if an element/attribute is required, the document should contain that element/attribute).

2 val.

- (b) Write a valid XML document, that conforms to the XSD, that is minimal (*i.e.* if an element can appear a minimum of one time, the document should only contain that element one time; if an element/attribute is optional, the document should not contain that element/attribute).

5. Consider the following XML documents:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <products>
3   <product name="A" id="123">
4     <part id="X3" count="3" />
5     <part id="Y2" count="1" />
6     <part id="Z6" count="2" />
7   </product>
8   <product name="B" id="456">
9     <part id="G3" count="1" />
10    <part id="Y6" count="2" />
11  </product>
12  <product name="C" id="789">
13    <part id="K9" count="6" />
14    <part id="Y6" count="3" />
15  </product>
16 </products>
```

Listing 4: products1.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <list>
3   <product>
4     <name>A</name>
5     <parts>
6       <part count="3">X3</part>
7       <part count="1">Y2</part>
8       <part count="2">Z6</part>
9     </parts>
10  </product>
11  <product>
12    <name>B</name>
13    <parts>
14      <part count="1">G3</part>
15      <part count="2">Y6</part>
16    </parts>
17  </product>
18  <product>
19    <name>C</name>
20    <parts>
21      <part count="6">K9</part>
22      <part count="3">Y6</part>
23    </parts>
24  </product>
25 </list>
```

Listing 5: products2.xml

2 val.

- (a) Write an XPATH expression that, when applied to the document *products2.xml*, returns the name of all products that have a part with a count greater than 5.

3 val.

- (b) Write an XSL transformation that converts the document exemplified in *products1.xml* into a document as exemplified in *products2.xml*.