Web Languages and Technologies

Faculdade de Engenharia da Universidade do Porto
12th January 2015

Duration: 2h / With Consultation

Name: _____

Number: _____

1. Consider the following HTML code:

```
1  <article class="post" id="first">
2    <header>
3        <h1>Title</h1>
4    </header>
5    <p>First paragraph</p>
6    <p>Second paragraph</p>
7    <footer>
8        <p>This is a footer</p>
9    </footer>
10 </article>
```

And the following CSS code:

```
1  article p {color: red;}              /* R1 */
2  article > p {color: blue;}           /* R2 */
3  .post p:first-child {color: green;} /* R3 */
4
5  p + p {color: yellow;}               /* R4 */
6  #first footer {color: magenta;}      /* R5 */
7  .post {color: cyan;}                 /* R6 */
```

1½ val. (a) Calculate the specificity of each one of the rules:

R1	R2	R3	R4	R5	R6
(0,0,0,2)	(0,0,0,2)	(0,0,2,1)	(0,0,0,2)	(0,1,0,1)	(0,0,1,0)

1 val. (b) Taking into consideration only the rules **R1 to R3**, indicate the color of each one of the texts in the page:

Title	1st Par	2nd Par	Footer
inherit	blue	blue	green

1 val. (c) Taking into consideration **all** the rules, indicate the color of each one of the texts in the page:

Title	1st Par	2nd Par	Footer
cyan	blue	yellow	green

2. Consider the following *string*: `Peter Piper picked a peck of pickled peppers.`

   For each one of the regular expressions shown below, underline the first match:

½ val. (a) `/pi.*ck/`
   `Peter Piper picked a peck of pickled peppers.`

½ val. (b) `/[a-k]{3}/`
   `Peter Piper picked a peck of pickled peppers.`

½ val. (c) `/(\w{4}).\1/+`
   `Peter Piper picked a peck of pickled peppers.`

½ val. (d) `/ck\b/`
   `Peter Piper picked a peck of pickled peppers.`

½ val. (e) `/(pick|peck)(?=l)/`
   `Peter Piper picked a peck of pickled peppers.`

½ val. (f) `/(?<=pep)per/`
   `Peter Piper picked a peck of pickled peppers.`

3. Consider the following HTML code excerpt:

```
1 <input name="color" type="text" value="#336699">
2 <button id="copy" value="Copy">
3 <button id="send" value="Send">
4 <div class="box"></div>
```

   Also consider that the complete page can have other *input*, *button* and *div* elements. Write the *jQuery* code needed so that:

1 val. (a) When the *copy* button is pressed, the background color of the *div* changes into the color specified in the *input* value.

Name: _____

Number: _____

2 val.

(b) When the *send* button is pressed, the background color of the *div* is sent, in a variable called *color*, as an *Ajax* request to the address *http://www.coloranalyzer.com/*. The *div* text should change into the result of the request. Considerer that the result, in JSON, has the following format: `{"result": "good"}`.

(Continues in the other side...)

2½ val. 4. Create a well-formed and valid XML document according to the following XSD:

```
<?xml version="1.1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="car">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="plate" type="plate"/>
        <xs:element name="make" type="xs:string"/>
        <xs:element name="model" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="age" use="required" type="xs:number"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="cars">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="car" minOccurs="2" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="plate">
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{3}-[A-Z]{3}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```