

Web Languages and Technologies

Faculdade de Engenharia da Universidade do Porto
19th January 2016

Duration: 2h / With Consultation

Name: _____

Number: _____

1. Consider the following HTML code:

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

And the following CSS code:

```
1 #first div {color: blue;} /* R1 */
2 div .header {color: magenta;} /* R2 */
3 p + p + div :first-child {color: red;} /* R3 */
4
5 h1 {color: green;} /* R4 */
6 div + p {color: cyan;} /* R5 */
7 p {color: yellow;} /* R6 */
```

1½ val.

(a) Calculate the specificity of each one of the rules:

R1	R2	R3	R4	R5	R6

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

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

2. Consider the following *string*: When you write copy you have the right to copyright the copy you write
For each one of the regular expressions shown below, underline the first match:

1/2 val.

(a) /copy.*right/

When you write copy you have the right to copyright the copy you write

1/2 val.

(b) /[write]/

When you write copy you have the right to copyright the copy you write

1/2 val.

(c) /(\w{4}).*\1/

When you write copy you have the right to copyright the copy you write

1/2 val.

(d) /write\$/

When you write copy you have the right to copyright the copy you write

1/2 val.

(e) /(ri|py)(?!t)/

When you write copy you have the right to copyright the copy you write

1/2 val.

(f) /(\w{3,})?.*\1/

When you write copy you have the right to copyright the copy you write

3. Consider the following HTML code excerpt:

```
1 <form id="register" action="register.php" method="post">
2   <input name="username" type="text">
3   <input name="password" type="password">
4   <input type="submit" value="Register">
5 </form>
```

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

1 val.

(a) When the *password input* loses focus, it is verified if it contains at least 8 characters with at least one of them being a symbol other than a letter, a number or an underscore. If that's not the case, the input's border should become red.

Name: _____

Number: _____

2 val.

- (b) When the *submit* button is clicked, the value of the *username input* should be sent, inside a variable named *username*, in an *Ajax POST* request to the address *verifyusername.php*. If the response indicates that the username is not valid, the *border* of the input should become red and the form should not be submitted. Consider that the result, in JSON format, can be either `{"valid": "true"}` or `{"valid": "false"}`.



(Continues on the other side...)

2½ val.

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

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:complexType name="productType">
4     <xs:sequence>
5       <xs:element name="name" type="xs:string"/>
6       <xs:element name="price" type="xs:decimal"/>
7     </xs:sequence>
8     <xs:attribute name="id" type="xs:integer"/>
9     <xs:attribute name="qty" type="xs:integer"/>
10  </xs:complexType>
11  <xs:complexType name="orderType">
12    <xs:sequence>
13      <xs:element name="product" type="productType" maxOccurs="unbounded" minOccurs="3"/>
14      <xs:element name="wrap" type="xs:integer" maxOccurs="unbounded" minOccurs="2"/>
15    </xs:sequence>
16    <xs:attribute name="number" type="xs:integer"/>
17  </xs:complexType>
18  <xs:element name="order" type="orderType">
19    <xs:key name="productKey">
20      <xs:selector xpath="product"/>
21      <xs:field xpath="@id" />
22    </xs:key>
23    <xs:keyref name="productRef" refer="productKey">
24      <xs:selector xpath="wrap"/>
25      <xs:field xpath="."/>
26    </xs:keyref>
27  </xs:element>
28 </xs:schema>
```

