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
(0,0,1,1)	(0,1,0,1)	(0,1,0,3)	(0,0,0,1)	(0,0,0,2)	(0,0,0,1)

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
magenta	inherit	inherit	red

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
green	cyan	yellow	red

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:

½ val. (a) `/copy.*right/`

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

½ val. (b) `/[write]/`

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

½ val. (c) `/(\w{4}).*\1/`

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

½ val. (d) `/write$/`

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

½ val. (e) `/(ri|py)(?!t)/`

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

½ 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.

```
let pass = document.getElementsByName('password')[0];
pass.addEventListener('blur', function(){
  if(!(/\W{1,}/.test(pass.value) && pass.value.length >= 8))
  pass.style.borderColor = "red";
  else pass.style.borderColor = "initial";
})
```

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"}`.

```
let form = document.querySelector('body form#register');
let username = form.children.username;
form.onsubmit = function(e){
   let request = new XMLHttpRequest();
   request.open('POST', 'verifyusername.php', false);
   let reply = false;
   request.onload = function(data){
      reply = JSON.parse(data.target.response).valid == "true";
      if(reply)
        username.style.borderColor = "initial";
      else
        username.style.borderColor = "red";
   };
   request.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
   request.send('username=' + username.value);
   return reply;
};
```

(Continues on the other side...)

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="productType">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="price" type="xs:decimal"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:integer"/>
    <xs:attribute name="qty" type="xs:integer"/>
  </xs:complexType>
  <xs:complexType name="orderType">
    <xs:sequence>
     <xs:element name="product" type="productType" maxOccurs="unbounded" minOccurs="3"/>
     <xs:element name="wrap" type="xs:integer" maxOccurs="unbounded" minOccurs="2"/>
    </xs:sequence>
    <xs:attribute name="number" type="xs:integer"/>
  </xs:complexType>
  <xs:element name="order" type="orderType">
    <xs:key name="productKey">
      <xs:selector xpath="product"/>
      <xs:field xpath="@id" />
    </xs:key>
    <xs:keyref name="productRef" refer="productKey">
      <xs:selector xpath="wrap"/>
      <xs:field xpath="."/>
    </xs:keyref>
  </xs:element>
</xs:schema>
```