Web Languages and Technologies

Faculdade de Engenharia da Universidade do Porto
30th January 2015

Duration: 2h / With Consultation

Name: _____

Number: _____

1. Consider the following HTML code:

```
1  <div id="selection">
2    <p><a href="">Two Lists</a></p>
3    <ul>
4      <li class="selected">First</li>
5      <li>Second</li>
6    </ul>
7    <ol class="other">
8      <li>Third</li>
9    </ol>
10 </div>
```

And the following CSS code:

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

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

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

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

Two Lists	1st First	2nd Second	Third
default anchor color	blue	green	yellow

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

Two Lists	First	Second	Third
default anchor color	cyan	green	cyan

2. Consider the following *string*: `The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

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

½ val. (a) `/led.+ro/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

½ val. (b) `/[thir]+[^e]/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

½ val. (c) `/(\w{3}.+\1)/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

½ val. (d) `/ll.*e\b/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

½ val. (e) `/(h|r|t){3}/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

½ val. (f) `/(?<!h)o(?=u)/`
`The thirty-three thieves thought that they thrilled the throne throughout Thursday.`

3. Consider the following HTML code excerpt:

```
1 <script>
2 var secret = Math.floor((Math.random() * 100) + 1); // generates random number
3 var tries = 0;
4 </script>
5 <input name="username" type="text" placeholder="username">
6 <input name="guess" type="text">
7 <input id="guess" type="button" value="Guess">
```

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

1 val. (a) When the *guess* button is pressed, if the value in the input named *guess* is lower than the variable *secret*, a dialog with the sentence "go up"should be shown, if it is higher, the sentence should read "go down"and if they are the same, a function named *correct* should be called. The *tries* variable should be increased by one in any of the cases.

```
let guess = document.querySelector('input[name="guess"]');
let guessButton = document.querySelector('input#guess[type="button"]');
guessButton.addEventListener('click', function (event){
 tries++;
 if(guess.value > secret){
        alert('Go Down');
      } else if(guess.value < secret){
        alert('Go Up');
      } else {
        correct();
      tries = 0;
      }
});
```

Name: _____

Number: _____

2 val.

(b) Create the function named *correct*, that was mentioned in the previous question, so that it shows a dialog with the sentence "correct"and also makes an *Ajax* call to a *save_score.php* script. The username (input with name *username*) and the number of tries (variable *tries*) should also be sent to that script. Inform the user if the script was called successfully or not.

```
function correct(){
 alert("correct");
  let request = new XMLHttpRequest();
  let username = document.querySelector('input[name="username"]').value;
  let requestString = "username=" + username + "&tries=" + tries;
  request.open('GET', 'save_score.php?' + requestString);
  request.onload = function(data){
   if(data.target.status == 200)
       alert('success');
    else
       alert('unsuccess');
  };
  request.send();
}
```

(Continues on 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="student">
    <xs:complexType>
      <xs:choice>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="nickname" type="xs:string"/>
      </xs:choice>
      <xs:attribute name="code" type="scode"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="students">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="student" minOccurs="2" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="count" type="xs:integer"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="scode">
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{5}[A-Z]{2}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```