

# Linguagens e Tecnologias Web

Faculdade de Engenharia da Universidade do Porto  
12 de Janeiro de 2015

Duração: 2h / Com Consulta

Nome: \_\_\_\_\_

Número: \_\_\_\_\_

1. Considere o seguinte código HTML:

```
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>
```

E o seguinte código CSS:

```
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) Indique a especificidade de cada uma das regras:

R1	R2	R3	R4	R5	R6

1 val.

(b) Considerando apenas as regras de **R1 a R3**, indique a cor de cada um dos textos:

Title	1st Par	2nd Par	Footer

1 val.

(c) Considerando **todas as regras**, indique a cor de cada um dos textos:

Title	1st Par	2nd Par	Footer

2. Considere a seguinte *string*: Peter Piper picked a peck of pickled peppers.

Para cada uma das expressões regulares apresentadas de seguida, sublinhe qual o primeiro *match*:

$\frac{1}{2}$  val.

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

$\frac{1}{2}$  val.

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

$\frac{1}{2}$  val.

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

$\frac{1}{2}$  val.

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

$\frac{1}{2}$  val.

(e) `/(pick|peck)(?=1)/`  
Peter Piper picked a peck of pickled peppers.

$\frac{1}{2}$  val.

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

3. Considere o seguinte excerto HTML:

```
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>
```

Considere que pode haver outros elementos *input*, *button* e *div* no documento. Escreva o código *jQuery* necessário para que:

1 val.

(a) Quando o botão *copy* é clicado, a cor de fundo do *div* mude para a cor especificada no texto do *input*.

Nome: \_\_\_\_\_

Número: \_\_\_\_\_

2 val.

- (b) Quando o botão *send* é clicado, a cor de fundo do *div* seja enviada, numa variável *color*, num pedido *Ajax* para o endereço *http://www.coloranalyzer.com/*. Mude o texto do *div* para o resultado do pedido. Considere que o resultado, em JSON, vem no seguinte formato: `{"result": "good"}`.



(Continua do outro lado...)

2½ val.

4. Crie um documento XML que seja bem formado e válido segundo o seguinte XSD:

```
1 <?xml version="1.1"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="carro">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="matricula" type="matricula"/>
7         <xs:element name="marca" type="xs:string"/>
8         <xs:element name="modelo" type="xs:string"/>
9       </xs:sequence>
10      <xs:attribute name="idade" use="required" type="xs:number"/>
11    </xs:complexType>
12  </xs:element>
13  <xs:element name="carros">
14    <xs:complexType>
15      <xs:sequence>
16        <xs:element ref="carro" minOccurs="2" maxOccurs="unbounded"/>
17      </xs:sequence>
18    </xs:complexType>
19  </xs:element>
20  <xs:simpleType name="matricula">
21    <xs:restriction base="xs:string">
22      <xs:pattern value="\d{3}-[A-Z]{3}"/>
23    </xs:restriction>
24  </xs:simpleType>
25 </xs:schema>
```

