

Linguagens e Tecnologias Web

Faculdade de Engenharia da Universidade do Porto
30 de Janeiro de 2013

Duração: 2h

Com consulta de apontamentos em papel

1. Dados os excertos de HTML e CSS abaixo apresentados:

```
1 <body>
2   <div id="a1" class="a2 x">A</div>
3   <div id="b1" class="b2 x">B</div>
4   <div id="c1" class="c2 y">C</div>
5   <div id="d1" class="d2 y">D</div>
6 </body>
```

Listagem 1: Excerto HTML

```
1 html, body {margin: 0px;}
2
3 div {
4   border: 1px solid black;
5   width: 100px;
6   height: 100px;
7   text-align: center;
8   background-color: cyan;
9 }
10
11 #a1 { position: absolute; top: 10px; left: 10px; }
12 #b1 { position: relative; top: 10px; left: 120px; }
13 #c1 { margin: 10px; }
14 #d1 { margin: 10px; }
15
16 *, div#a1 {background-color: yellow; } /* R1 */
17 body>div, body>div>#c1 { background-color: red; } /* R2 */
18 body>div.c2, body>* { background-color: blue; } /* R3 */
19 .c2+*, #a1+.x { background-color: green; } /* R4 */
```

Listagem 2: Excerto CSS

1 val.

(a) Calcule a especificidade de cada um dos seletores das regras R1 a R4.

1 val.

(b) Desenhe o resultado aproximado que seria apresentado num *browser* sem contar com as 4 regras marcadas de R1 a R4. Não se esqueça de indicar as distâncias entre *divs* e de desenhar o bordo da janela.

2 val.

(c) Indique a cor de fundo com que ficaria cada *div* se as regras marcadas de R1 a R4 fossem introduzidas uma de cada vez (*i.e.* primeiro apenas com a regra R1, depois com as regras R1 e R2, ...).

2. Escreva um trecho de código *jQuery* que, quando aplicado ao código da pergunta anterior, produza o seguinte resultado:

2¹/₂ val.

(a) *Clicando* em qualquer um dos *divs*, deve ser criado um novo *div* vazio com a mesma cor de fundo do *div clicado*. Este novo *div* deve ser acrescentado como último elemento do *body*. Deve ser ainda efetuado um pedido *Ajax* ao URL `http://getrandomtext.com/`. Ao receber a resposta, deverá mudar o texto do novo *div* para o texto recebido. Considere que a resposta vem codificada em JSON com o seguinte formato:

```

1 | {
2 |   "rid": 362147,
3 |   "text": "Some Random Text"
4 | }

```

1½ val.

(b) Quando o rato entra num *div* (incluindo os que foram criados dinamicamente na alínea anterior), este deve ficar com a cor de fundo do *body* e o *body* deve ficar com a cor de fundo do *div*. Ou seja, o *div* e o *body* devem trocar de cores entre si.

3. Considere que um ficheiro contém uma lista de nomes completos de pessoas (um nome por linha). Cada nome é composto por várias palavras separadas por *whitespaces*. Cada palavra pode apenas conter letras de 'a' a 'z' (minúsculas ou maiúsculas).

Escreva uma expressão regular que faça *match* com os nomes (completos) que obedeçam às seguintes condições:

1 val.

(a) A primeira palavra é Tiago e a última é Silva.

1 val.

(b) A primeira palavra é Tiago e o nome não contém a palavra Silva.

1 val.

(c) A primeira palavra começa com a mesma letra que a última palavra.

1 val.

(d) O nome tem duas palavras iguais.

3 val.

4. Considere que o seguinte documento XML se encontra guardado num ficheiro chamado *favorites.xml*:

```

1 | <?xml version="1.0" encoding="UTF-8" ?>
2 | <favorites>
3 |   <games>
4 |     <game title="Call of duty" esrb="18" code="0001">
5 |       <genre name="First Person Shooter" code="FPS"/>
6 |     </game>
7 |     <game title="Pro Evolution Soccer" esrb="4" code="0002">
8 |       <genre name="Sports" code="SPO"/>
9 |       <genre name="Simulation" code="SIM"/>
10 |      <genre name="Arcade" code="ARC"/>
11 |    </game>
12 |    <game title="Painkiller" esrb="18" code="0003">
13 |      <genre name="First Person Shooter" code="FPS"/>
14 |    </game>
15 |  </games>
16 |  <sites>
17 |    <site title="SiFEUP" url="http://www.fe.up.pt">
18 |      <info type="education" />
19 |    </site>
20 |    <site title="Google" url="http://www.google.com">
21 |      <info type="everything" />
22 |    </site>
23 |    <site title="Wordpress" url="http://www.wordpress.com">
24 |      <blog>
25 |        <author name="John Doe" />
26 |        <author name="Bill Doe" />
27 |      </blog>
28 |    </site>
29 |  </sites>
30 | </favorites>

```

Listagem 3: Documento XML

Escreva o XSD utilizado para validar o XML apresentado acima.

O título, índice ESRB e código de um jogo são atributos obrigatórios, assim como o título e URL de um site.

Considere que cada jogo tem que ter pelo menos um género. Os atributos *name* e *code* do género são obrigatórios. O código de um jogo é sempre um número com 4 dígitos e não pode ter valores repetidos.

Um site pode ser informativo ou um *blog*, devendo para tal conter os elementos *info* ou *blog* (apenas um destes). O elemento *info* pode conter, opcionalmente, um atributo *type*. O elemento *blog* contém uma lista com pelo menos um autor. O nome do autor é obrigatório.

5. Considere o seguinte documento XML:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project>
3   <task name="requirements">
4     <assigned person="2" />
5     <assigned person="3" />
6   </task>
7   <task name="development">
8     <assigned person="1" />
9     <assigned person="2" />
10  </task>
11  <task name="testing">
12    <assigned person="1" />
13    <assigned person="2" />
14    <assigned person="3" />
15  </task>
16  <person id="1">
17    <name>Miguel</name>
18    <email>miguel@email.com</email>
19  </person>
20  <person id="2">
21    <name>Raquel</name>
22    <email>raquel@email.com</email>
23  </person>
24  <person id="3">
25    <name>Carlos</name>
26    <email>carlos@email.com</email>
27  </person>
28 </project>
```

Listagem 4: project.xml

E a seguinte transformação XSL:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3   <xsl:template match="/*">
4     <persons>
5       <xsl:apply-templates select="/*[@id]"/>
6     </persons>
7   </xsl:template>
8   <xsl:template match="person">
9     <person email="{email}">
10      <xsl:attribute name="name">
11        <xsl:value-of select="name"/>
12      </xsl:attribute>
13      <xsl:for-each select="/project/task[assigned/@person=current()/@id]">
14        <task name="{@name}" />
15      </xsl:for-each>
16    </person>
17  </xsl:template>
18 </xsl:stylesheet>
```

Listagem 5: transform.xml

2 val.

(a) Escreva uma expressão XPATH que, quando aplicada ao documento *project.xml*, obtenha o nome de todas as tarefas com mais do que duas pessoas.

3 val.

(b) Apresente o resultado da aplicação do XSL *transform.xml* sobre o documento *project.xml*.

Nota: A função *current()* retorna o nodo que está a ser processado.