

Linguagens e Tecnologias Web

Faculdade de Engenharia da Universidade do Porto
19 de Janeiro de 2016

Duração: 2h / Sem Consulta

Nome: _____

Número: _____

- 1 val. 1. Em que situações se deve usar o elemento HTML *aside*?
- ☐ Quando o elemento vai ficar num dos lados da página.
 - ☐ Quando o elemento não faz parte do conteúdo principal da página.
 - ☐ Quando contém elementos de navegação.
 - ☐ Quando não contém um *header*.
- 1 val. 2. Qual a forma correcta de prevenir um ataque to tipo *Cross-Site Request Forgery*?
- ☐ Usar sempre formulários do tipo POST.
 - ☐ Fazer os pedidos usando AJAX.
 - ☐ Usar um *token* de segurança diferente para cada sessão.
 - ☐ Ter sempre um diálogo de confirmação.
- 1 val. 3. Uma password deve ser sempre cifrada usando um algoritmo de *hashing* seguro. Que tipo de ataque é prevenido ao usar uma função de *hash* lenta?
- ☐ SQL Injection
 - ☐ Dictionary Attack
 - ☐ Brute Force
 - ☐ Rainbow Tables
- 1 val. 4. No protocolo HTTP, como são enviadas as *cookies*?
- ☐ Numa resposta AJAX usando JSON.
 - ☐ Na barra de endereços do *browser*.
 - ☐ Ficam guardadas no servidor.
 - ☐ Num header HTTP.
- 1 val. 5. Em PHP, ao declararmos uma variável como *global* dentro de uma função, essa variável:
- ☐ Fica disponível em qualquer outra função.
 - ☐ Fica disponível em qualquer outro script dentro da directoria.
 - ☐ Refere-se a uma variável global do *script*.
 - ☐ Não pode ser usada noutra função.

(continua do outro lado...)

1 val.

6. Assuma que o seguinte código *jQuery* se encontra definido dentro do evento *ready* de um documento HTML. Analise, com atenção, o código e indique qual o comportamento esperado:

```
1 | $('a').click(say('Hello'));  
2 |  
3 | function say(message) {  
4 |     alert(message);  
5 | }
```

- ☐ É mostrado um alerta com a mensagem *Hello* assim que a página carrega.
- ☐ É mostrado um alerta com a mensagem *Hello* quando se carrega num link.
- ☐ Não acontece nada.
- ☐ A página não carrega.

1 val.

7. Explique sucintamente quais nodos são seleccionados pela seguinte expressão *XPath*:

`./person/job`

1 val.

8. Em XSD, um *complexType* pode conter texto?

- ☐ Sim, desde que não contenha outros elementos XML.
- ☐ Sim, desde que o atributo *mixed* tenha o valor *true*.
- ☐ Sim, desde que o atributo *text* tenha o valor *true*.
- ☐ Não, só elementos *simpleType* podem conter texto.

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 <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>
```

E o seguinte código CSS:

```
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) Indique a especificidade de cada uma das regras (e.g. 0,2,2,1):

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*:

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

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

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

(a) `/copy.*right/`

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

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

(b) `/[write]/`

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

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

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

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

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

(d) `/write$/`

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

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

(e) `/(\r|\py)(?!t)/`

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

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

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

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

3. Considere o seguinte excerto HTML:

```
1 <form 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>
```

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

1 val.

(a) Quando o *input password* perde o foco, se verifique se esta tem pelo menos 8 caracteres e que pelo menos um deles seja um símbolo que não uma letra, um número ou um *underscore*. Caso tal não se verifique, o *border* do input deve passar a vermelho.

Nome: _____

Número: _____

2 val.

- (b) Quando o botão de *submit* seja *clicado*, o valor do *input username* seja enviado, numa variável *username*, num pedido *Ajax* do tipo *POST* para o endereço *verifyusername.php*. Caso a resposta indique que o *username* não é válido, o *border* do *input* deve passar a vermelho e o formulário não deve ser submetido. Considere que o resultado, em JSON, vem no seguinte formato: `{"valid": "true"}` ou `{"valid": "false"}`.



(Continua do outro lado...)

4. Crie um documento XML que seja bem formado e válido segundo o seguinte 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>
```