

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 do 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`

Todos os nodos job filhos de qualquer nódo person existente

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.

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*:

1/2 val.

(a) /copy.*right/

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

1/2 val.

(b) /[write]/

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

1/2 val.

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

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

1/2 val.

(d) /write\$/

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

1/2 val.

(e) /(ri|py)(?!t)/

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

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 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>
```

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.

```
$('#register input[name="password"]').focusout(function() {
    if (this.value.length < 8 || /[0-9-]/.test(this.value) == false)
        $('#register input[name="password"]').css('border-color', 'red')
})
```

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"}*.

```
$("#register input[type="submit"]').click(function() {  
    let username = $('#register input[name="username"]').val()  
    let request = new XMLHttpRequest()  
    request.open('POST', 'verifyusername.php', false)  
    request.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')  
    request.addEventListener('load', function() {  
        let response = JSON.parse(this.responseText)  
        if (response.valid == "false") {  
            $('#register input[name="username"]').css('border-color', 'red')  
            document.querySelectorAll('#register').preventDefault()  
        }  
    })  
})
```

}

```
request.send(encodeForAjax('username': username))
```

}

(Continua do outro lado...)