

Faculdade de Engenharia da Universidade do Porto
1 de Fevereiro de 2016

Duração: 2h / Sem Consulta

Nome: _____

Número: _____

1 val.

1. Que atributo é usado para definir que *inputs* do tipo radio pertencem ao mesmo grupo?

- ☐ O atributo *group*.
- ☒ O atributo *name*.
- ☐ O atributo *label*.
- ☐ O atributo *id*.



1 val.

2. Os codificadores dependentes do contexto (*context aware encoders*) são usados quando queremos fazer *output* de dados em que não confiamos. Que tipo de ataque tentam prevenir?

- ☐ CSRF (*Cross-site Request Forgery*).
- ☐ SQL Injection.
- ☒ XSS (*Cross-site Scripting*).
- ☐ Path Traversal Attack.



1 val.

3. Quando nos ligamos a um servidor usando o protocolo HTTPS, é usada um esquema de chave pública e chave privada. Como é que se garante que a chave pública transmitida pelo servidor é realmente desse servidor e que não estamos a sofrer um man in the middle attack?

- ☐ A chave foi obtida através de um servidor de chaves seguro.
- ☐ As chaves públicas contêm um *checksum* que permite verificar a sua validade.
- ☒ A chave foi assinada por uma entidade certificadora reconhecida pelo *browser*.
- ☒ A chave é transmitida através de um mecanismo seguro (TLS).



1 val.

4. No protocolo HTTP, qual dos seguintes métodos **não** é idempotente?

- ☐ GET
- ☐ DELETE
- ☒ POST
- ☐ HEAD



1 val.

5. O que é impresso pelo seguinte código PHP:

```
1 $a = array('a', 'b', 'c');  
2 foreach ($a as $b => $c)  
3     echo $b;
```



012

(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 | alert($('a').css('color'));
```

- ☐ Aparece uma caixa de diálogo com a cor de todos os *links* do documento.
- ☐ Aparece uma caixa de diálogo para cada *link* do documento com a sua cor.
- ☒ Aparece uma caixa de diálogo com a cor do primeiro *link* do documento.
- ☐ Aparece um erro na consola pois não podemos fazer *alert* de um *array*.

- 1 val. 7. Em XPath, qual é o eixo por omissão?

- ☐ element
- ☐ parent
- ☐ comment
- ☒ child



- 1 val. 8. Em XSD, o que significa o atributo *fixed* no seguinte código?

```
1 | <xs:attribute name="value" type="xs:integer" fixed="5"/>
```

- ☒ O atributo tem de estar sempre presente e o valor tem de ser sempre 5.
- ☐ O valor do atributo é 5 por omissão.
- ☒ Se o atributo estiver presente, tem de ter o valor 5.
- ☐ O atributo tem de aparecer 5 vezes.



3. Considere o seguinte excerto HTML que representa um teclado virtual que pretende impedir ataques usando *keyboard loggers*:

```
1 <form id="pin" method="post">
2   <input type="text" name="username">
3   <input type="text" name="pin">
4   <input type="submit" value="Verify">
5 </form>
6 <div id="keypad">
7   <a href="#">1</a> <a href="#">2</a> <a href="#">3</a><br>
8   <a href="#">4</a> <a href="#">5</a> <a href="#">6</a><br>
9   <a href="#">7</a> <a href="#">8</a> <a href="#">9</a><br>
10 </div>
```

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

1 val.

- (a) Quando o utilizador *clique* num dos números do teclado virtual, o valor desse número seja acrescentado ao valor do *input* com o nome *pin*.

```
$('a').click(function() {
    $('input[name="pin"]').val($('input[name="pin"]').val() + this.text)
})
```

2 val.

- (b) Quando o botão de *submit* seja *clicado*, os valores dos *inputs* *username* e *pin* sejam enviados, em duas variáveis com nome *username* e *pin*, num pedido *Ajax* do tipo *POST* para o endereço *verify_pin.php*. Caso a resposta indique que o *pin* não é válido, o *border* do input *pin* deve passar a vermelho e o seu valor deve ser apagado. Considere que o resultado, em *JSON*, vem no seguinte formato: `{"valid": "true"}` ou `{"valid": "false"}`.

```

$( 'input [ type = "submit" ] ' ). click ( function () {
    let request = new XMLHttpRequest ()
    request . open ( 'POST' , 'verify_pin.php' , true )
    request . setRequestHeader ( 'Content-Type' , 'application/x-www-form-urlencoded' )
    request . addEventListener ( 'load' , function () {
        let response = JSON . parse ( this . responseText )
        if ( response . valid == "false" ) {
            $( 'input [ name = "pin" ] ' ) . css ( "border-color" , "red" )
            $( 'input [ name = "pin" ] ' ) . val ( " " )
        }
    } )
    let username = document . querySelector ( 'input ' ) . value
    let pin = document . querySelectorAll ( 'input ' ) [ 1 ] . value
    request . send ( encodeForAjax ( { 'username' : username , 'pin' : pin } ) )
} )

```

(Continua do outro lado...)