

Nome: _____

Número: _____

Nota: Nas perguntas de escolha múltipla, cada resposta errada desconta $\frac{1}{3}$ da cotação mas pode escolher mais do que uma opção.

1 val. 1. Para que dois *inputs* do tipo *radio* sejam considerados do mesmo grupo, é necessário que...

- ☐ Estejam em linhas consecutivas.
- ☒ Tenham o mesmo valor no atributo *name*.
- ☐ Tenham o mesmo valor no atributo *value*.
- ☐ Tenham o mesmo valor no atributo *group*.



1 val. 2. Que selector CSS se deve usar para seleccionar o título do primeiro artigo com classe featured?

- ☐ `article.featured :first-child h2 { ... }.`
- ☐ `article.featured h2:first-child { ... }.`
- ☐ `article:first-child .featured h2 { ... }.`
- ☒ `article.featured:first-child h2 { ... }.`

↳ selecciona o primeiro artigo de classe featured

1 val. 3. Um ataque do tipo *Session Hijacking* consiste em...

- ☐ Alterar a cookie de um utilizador usando CSRF.
- ☐ Forçar uma certa cookie de sessão a um utilizador.
- ☒ Obter uma cookie de sessão enviada de forma insegura.
- ☐ Tirar partido de cookies guardadas no browser sem serem cifradas.

? não tenho a certeza qual a correta

1 val. 4. Uma password cifrada sem salt mas com um algoritmo criptograficamente seguro é virtualmente impossível de descobrir?

- ☒ Só se a password for grande e complexa.
- ☐ Nunca é possível de descobrir. ✗
- ☐ É sempre fácil de descobrir. ✗
- ☐ Só se o algoritmo usado não for conhecido. ✗

1 val. 5. Ao navegar para um site cujo URL começa por *https://* temos sempre a certeza de que...

- ☒ Não está a existir um ataque do tipo *Man in the Middle*.
- ☐ O servidor usa a versão 2.0 do protocolo HTTP.
- ☐ A comunicação é cifrada.
- ☐ A versão do HTML é superior a 4.



(continua do outro lado...)

- 1 val. 6. Em PHP, a função *strpos* retorna a posição, começando em 0, da primeira ocorrência de uma *string* dentro de outra *string*; ou *false* se não a encontrar. Após o seguinte código, qual a forma correcta de verificar se nenhuma ocorrência foi encontrada?

```
$pos = strpos($search, $test);
```

- ☒ if (\$pos == false) echo "Not found";
- ☐ if (!\$pos) echo "Not found";
- ☐ if (\$pos === false) echo "Not found";
- ☐ if (\$pos != true) echo "Not found";

Necessário '===' porque 0 pode ser identificado como false usando '=='

- 1 val. 7. Em Javascript, uma função é...

- ☐ Um protótipo.
- ☐ Uma classe.
- ☐ Um contexto.
- ☒ Um objecto.



- 1 val. 8. Em XPath, qual é o eixo por omissão?

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

2. Considere a seguinte *string*:

Washing the washing machine while watching the washing machine washing washing

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

- $\frac{1}{2}$ val. (a) `/w.*[a-z]/`
Washing the washing machine while watching the washing machine washing washing
- $\frac{1}{2}$ val. ✓ (b) `/a[~s]/`
Washing the washing machine while watching the washing machine washing washing
- $\frac{1}{2}$ val. (c) `/([a-z]{3}).*?\1/`
Washing the washing machine while watching the washing machine washing washing
- $\frac{1}{2}$ val. ✓ (d) `/^.*?$/`
Washing the washing machine while watching the washing machine washing washing
- $\frac{1}{2}$ val. ✓ (e) `/(?<!the)washing/`
Washing the washing machine while watching the washing machine washing washing
- $\frac{1}{2}$ val. ✓ (f) `/(?:w)(a).*\1/`
Washing the washing machine while watching the washing machine washing washing

3. Considere o seguinte excerto HTML:

```
1 <div id="photos">
2   </li>
3   <ul>
4     <li></li>
5     <li></li>
6     <li></li>
7   </ul>
8   <a href="#" class="load">Load More</a>
9 </div>
```

Considere que pode haver outros elementos *ul*, *li*, *img* e *a* no documento, e que a lista de imagens pode ser diferente. Escreva o código *jQuery* necessário para que:

- 1 val. (a) Quando o utilizador *clique* numa imagem da lista, a *src* da imagem com a class *large* passe a ser a *src* dessa imagem com o prefixo *large/*.

```
$('li img').click(function() {
    $('#photos img').attr("src", "large/" + $(this).attr("src"));
});
```

2 val.

- (b) Quando o *link* com a classe *load* seja *clicado*, seja feito, um pedido *Ajax* do tipo *GET* para o endereço *getrandomimages.php*.

Quando o resultado desse pedido for recebido, devem ser acrescentadas à lista novas imagens com os endereços contidos no resultado. O resultado será sempre um array JSON com o formato usado no seguinte exemplo:

```
[ "horse.png", "cow.png", "pig.png" ]
```

```
$(a.load).click(function() {
    let request = new XMLHttpRequest()
    request.open("get", "getrandomimages.php", true)
    request.addEventListener("load", function() {
        let responses = JSON.parse(this.responseText)
        for (let i = 0; i < responses.length; i++) {
            $('#div#photos ul').append("<b><img src='\" + responses[i] + \"' /></b>")
        }
    })
    request.send()
})
```

(Continua do outro lado...)

4. Considere o seguinte documento em XML:

```
1 <authors>
2   <author country="Spain" name="Miguel de Cervantes">
3     <book year="1605" type="Novel">Don Quixote</book>
4   </author>
5   <author country="England" name="William Shakespeare">
6     <book year="1599" type="Tragedy">Hamlet</book>
7     <book year="1606" type="Tragedy">Macbeth</book>
8   </author>
9   <author country="Russia" name="Leo Tolstoy">
10    <book year="1865" type="Novel">War and Peace</book>
11  </author>
12  <author country="Portugal" name="Jose Saramago">
13    <book year="1995" type="Novel">Ensaio sobre a Cegueira</book>
14    <book year="1997" type="Novel">Todos os Nomes</book>
15  </author>
16 </authors>
```

Considerando que o nodo de contexto é a raiz do documento, escreva as expressões XPath que seleccionam os seguintes elementos:

1/2 val.

(a) O título de todos os livros.

`/authors / author / book / text ()`

✓

1/2 val.

(b) O título dos livros escritos depois de 1900.

`/authors / author / book [@ year > 1900] / text ()`

✓

1/2 val.

(c) Os anos em que foram escritos livros por autores Ingleses.

`/authors / author [@ country = "England"] / book / @ year`

✓

1 val.

(d) O nome dos autores que escreveram livros do tipo *Novel*.

`/authors / author / book [@ type = "Novel"] / parent :: author / @ name`