

Nome: _____

Nº aluno: _____

Notas Importantes:

Deve assinalar apenas uma opção como sendo a correta e/ou mais completa.

Nas questões de **escolha múltipla**, cada resposta errada implica uma penalização na respetiva cotação da pergunta (25%). Se o aluno não responder não há lugar a qualquer penalização. Deve identificar a resposta correta através da especificação da respetiva letra no quadrado destinado para esse efeito.

As cotações foram definidas tendo por base 20 valores.

[0.3 Valores]

1. A implementação de uma *Single Page Application (SPA)*:

- A. permite uma gestão local (*cliente*) do conteúdo da página, em que sempre que o conteúdo é alterado é necessário efetuar o *reload* de uma nova página.
- B. permite uma gestão local (*cliente*) do conteúdo da página, como tal sem necessidade de efetuar o *reload* de uma nova página.
- C. impede a comunicação com o servidor.
- D. permite uma gestão remota (*servidor*) do conteúdo da página, em que sempre que o conteúdo é alterado é necessário efetuar o *reload* de uma nova página.

☐

[0.4 Valores]

2. No contexto das *Single Page Application (SPA)* a tecnologia *Javascript/Frameworks JS* :

- A. não é relevante, uma vez que é orientada para minimizar a interação com o utilizador.
- B. é crucial, uma vez que a gestão do conteúdo é baseada em scripts JS.
- C. é crucial, uma vez que a gestão do conteúdo é baseada em scripts CSS.
- D. não é relevante, tendo sido substituída pela tecnologia SASS.

☐

[0.3 Valores]

3. O conceito web 2.0 está diretamente relacionado com:

- A. a incorporação de técnicas (inteligência artificial, etc.) que permitam otimizar e extrair conhecimento (*knowledge discovery*) a partir dos dados existentes.
- B. a progressiva eliminação do conceito de *Software as a Service (SaaS)*.
- C. a diminuição do *user generated content*.
- D. o surgimento das redes sociais, a partilha de vídeos, a generalização do conceito de *Software as a Service (SaaS)*.

☐

[0.4 Valores]

4. A portabilidade entre *browsers*:

- A. deve ser sempre evitada.
- B. deve ser sempre um objetivo a atingir, nesta perspetiva os *CSS Resets* são irrelevantes.
- C. deve ser sempre um objetivo a atingir, nesta perspetiva os *CSS Resets* são relevantes.
- D. pode ser evitada através da utilização de um *CSS Reset*.

☐

[0.4 Valores]

5. O conceito *Web Mobile*:

- A. tem vindo progressivamente a perder importância, tendência que se vai agravar com o surgimento das redes 5G.
- B. só deve ser considerado no contexto de uma *Single Page Application* (SPA).
- C. só deve ser considerado no contexto de uma *Multi Page Application* (MPA).
- D. é fundamental no contexto do *front end web development*.

☐

[0.3 Valores]

6. O *Hyper Text Markup Language* (HTML) é uma linguagem de:

- A. programação destinada essencialmente à estruturação de conteúdos.
- B. programação destinada essencialmente à estruturação e formatação de conteúdos.
- C. marcação destinada exclusivamente à formatação de conteúdos.
- D. marcação destinada essencialmente à estruturação de conteúdos.

☐

[0.3 Valores]

7. Uma variável é utilizada para armazenar valores, neste contexto o HTML:

- A. não permite a declaração de variáveis.
- B. permite a declaração de variáveis.
- C. permite a declaração de variáveis através da *keyword variable*.
- D. permite a declaração de variáveis através da *keyword important!*.

☐

[0.4 Valores]

8. Na tecnologia HTML, uma *empty tag*:

- A. não admite atributos.
- B. pode admitir atributos, os quais são sempre especificados na *closing tag*.
- C. é sempre definida com atributos, os quais são especificados na *opening tag*.
- D. pode admitir atributos.

☐

[0.4 Valores]

9. De acordo com a última versão do HTML a não especificação das tags `<html>`, `<head>` e `<body>`:

- A. impede a correta interpretação pelo browser do código HTML.
- B. permite a correta interpretação pelo browser do código HTML.
- C. permite a correta interpretação pelo browser apenas do código CSS.
- D. impede a correta interpretação pelo browser apenas do código CSS.

☐

[0.4 Valores]

10. Os formulários constituem um importante elemento de interatividade com o utilizador, neste contexto a tag `<fieldset>` permite:

- A. a introdução direta de dados num campo numérico.
- B. a criação de um menu *drop-down*.
- C. o agrupamento de campos, melhorando dessa forma a estrutura do formulário.
- D. a criação de um campo de texto.

☐

[0.4 Valores]

11. No campo de texto de um formulário HTML o atributo *placeholder* representa:

- A. o valor do campo, se o campo não for preenchido é este o valor a enviar ao servidor.
- B. o valor final do campo.
- C. apenas uma ajuda ao preenchimento do campo, nunca estabelecendo o valor do campo.
- D. uma ajuda ao preenchimento do campo e simultaneamente o valor inicial do campo.

☐

[0.4 Valores]

12. No contexto do HTML, um *block level element*:

- A. expande-se de forma a ocupar o seu *container* e ignora a especificação das respetivas largura e altura.
- B. expande-se de forma a ocupar o seu *container*, comportamento que pode ser alterado através da especificação da respetiva largura.
- C. não se expande de forma a ocupar o *container*, comportamento que pode ser alterado através da especificação da respetiva largura.
- D. não se expande de forma a ocupar o seu *container* e ignora a especificação das respetivas largura e altura.

☐

[0.4 Valores]

13. A `<meta>` tag é especificada no:

- A. `<head>` sendo relevante para o *Search Engine Optimization (SEO)*.
- B. `<body>` sendo irrelevante para o *Search Engine Optimization (SEO)*.
- C. `<body>` sendo relevante para o *Search Engine Optimization (SEO)*.
- D. `<head>` sendo irrelevante para o *Search Engine Optimization (SEO)*.

☐

[0.4 Valores]

14. Numa estrutura HTML a navegação interna pode ser estabelecida tendo por base a *tag*:

- A. <link> e o atributo **href**, sendo o destino identificado com base no atributo **class**.
- B. <a> e o atributo **href**, sendo o destino identificado com base no atributo **class**.
- C. <a> e o atributo **href**, sendo o destino identificado com base no atributo **id**.
- D. <link> e o atributo **href**, sendo o destino identificado com base no atributo **id**.

☐

[0.4 Valores]

15. No contexto da aplicação da tecnologia *Cascading Style Sheets (CSS)* ao HTML, a ligação a um ficheiro externo (CSS) é implementada tendo por base a *tag*:

- A. <include>.
- B. <link>.
- C. <css>.
- D. <import>.

☐

[0.4 Valores]

16. No contexto da tecnologia CSS, os seletores de **id** utilizados para identificar campos de formulários, são formados tendo por base:

- A. # seguido pela designação do id.
- B. \$ seguido pela designação do id.
- C. • seguido pela designação do id.
- D. % seguido pela designação do id.

☐

[0.4 Valores]

17. No contexto das CSS, a seleção de elementos HTML pode ser efetuada com base em seletores de contexto os quais são formados da seguinte forma:

- A. **Contexto** Elemento_a_Formatar
- B. Elemento_a_Formatar **Contexto**
- C. **Contexto**.Elemento_a_Formatar
- D. Elemento_a_Formatar.**Contexto**

☐

Nota: **Contexto** representa o seletor que define o contexto

[0.4 Valores]

18. Na eventualidade de ocorrer um conflito de formatação tendo por base seletores diferentes, mas com a mesma especificidade, prevalece a:

- A. última regra CSS a ser definida.
- B. primeira regra CSS a ser definida.
- C. regra CSS estabelecida pelo seletor que possui um seletor de *id*.
- D. regra CSS estabelecida pelo seletor que possui um seletor de *class*.

☐

[0.4 Valores]

19. No contexto da tecnologia CSS, o *box model* é composto por três zonas distintas:

- A. *content area/padding area/border area*
- B. *content area/inner area/margin area*
- C. *content area/inner area/outer area*
- D. *content area/padding area/margin area*

☐

[0.4 Valores]

20. A atribuição à propriedade *display* do valor *inline {display:inline-block};*:

- A. não permite a especificação da largura e altura do elemento.
- B. permite a especificação da largura e altura do elemento.
- C. é equivalente a *{display:inline;}*.
- D. só pode ser aplicada a *list-items* para construção de elementos de navegação vertical.

☐

[0.4 Valores]

21. Os seletores *:first-child* e *:first-of-type*:

- A. são perfeitamente equivalentes, podendo ser utilizados de forma indiferenciada.
- B. apesar de poderem produzir resultados semelhantes possuem objetivos diferentes, sendo o *:first-of-type* um seletor mais imune às alterações na estrutura do HTML do que o seletor *:first-child*.
- C. apesar de poderem produzir resultados semelhantes possuem objetivos diferentes, sendo o *:first-child* um seletor mais imune às alterações na estrutura do HTML do que o seletor *:first-of-type*.
- D. são totalmente diferentes, nunca permitindo selecionar os mesmos elementos.

☐

22. Considerando a estrutura HTML representada na figura 2, defina uma regra CSS tal que permita definir:

```
<div>
  <p> Tecnologias <span>Web</span></p>
  <ul>
    <li>Estrutura</li>
    <li>Apresentação</li>
    <li>Comportamento</li>
  </ul>
</div>
<div>
  <p>Conteúdos <span>Teórica:</span></p>
  <ul>
    <li>HTML </li>
    <li>CSS </li>
    <li>JavaScript</li>
  </ul>
</div>
```

Figura 2 – Estrutura HTML

(*) : **não são consideradas válidas** as respostas que afetem mais elementos do que aqueles que são especificados.

[0.6 Valores]

22.1 a cor do texto vermelha para os primeiros (“Estrutura”; “HTML”) das duas listas.

[0.6 Valores]

22.2 a cor do texto laranja apenas* para a 2ª opção da lista (“CSS”) do ultimo <div>.

[0.6 Valores]

22.3 a cor do texto laranja e a dimensão da fonte a 2em apenas* para a primeira letra do parágrafo do segundo <div>.

[0.6 Valores]

22.4 a cor de fundo laranja apenas* para o do primeiro <div>.

23. Considere os seguintes códigos HTML e CSS, sendo que as propriedades são aplicadas sequencialmente:

<pre> 1 2 3 4 </pre>	<pre>ul{ display: flex; width:220px; list-style: none} li{ width:100px; border:1px solid; margin:2px}</pre>
---	--

Figura 3 – Código HTML/CSS

[0.4 Valores]

23.1. Represente graficamente o output do código acima representado

Nota: deve considerar os valores por defeito das propriedades *flex-wrap* e *flex-direction* que são respetivamente *nowrap* e *row*.

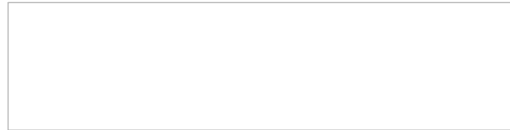
[0.4 Valores]

- 23.2. Represente graficamente o output do código acima representado considerando que foi adicionado ao código CSS a regra `ul{flex-wrap: wrap}`.



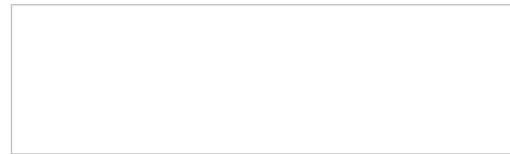
[0.4 Valores]

- 23.3. Represente graficamente o output do código acima representado considerando que foi adicionado ao código CSS a regra `ul{flex-direction: column-reverse}`.



[0.4 Valores]

- 23.4. Tirando partido das propriedades específicas dos *flex-items*, defina uma regra tal que o *list item* com o valor 3 surja representado na ultima posição (fundo da lista).



[0.4 Valores]

24. No contexto do CSS *flexbox*, a propriedade ***flex***:

- A. não existe.
- B. aplica-se ao *flex container*.
- C. aplica-se aos *flex items*.
- D. é equivalente à propriedade *flex-grow*.

☐

[0.4 Valores]

25. *CSS grid* aplica-se:

- A. exclusivamente a uma disposição unidimensional dos elementos, e baseia-se em dois conceitos chave: *grid container* e *grid item*.
- B. tipicamente a uma disposição bidimensional dos elementos, e baseia-se em dois conceitos chave: *grid container* e *grid item*.
- C. tipicamente a uma disposição bidimensional dos elementos, e baseia-se em dois conceitos chave: *grid section* e *grid wrap*.
- D. exclusivamente a uma disposição bidimensional dos elementos, e baseia-se em dois conceitos chave: *grid flex* e *grid element*.

☐

[0.4 Valores]

26. No contexto *CSS grid* as propriedades *grid-template-rows*, *grid-template-columns* :

- A. não se encontram definidas.
- B. definem apenas o número de linhas e colunas.
- C. definem o número de linhas e colunas assim como as respetivas dimensões.
- D. podem ser utilizadas desde que se garanta que a propriedade *display* possui o valor *flex*

☐

[0.4 Valores]

27. A propriedade *grid-template-areas* destina-se a definir:

- A. as dimensões das diferentes áreas da *CSS grid*.
- B. a designação das diferentes áreas da *CSS grid*, o que pode ser muito útil na atribuição dos conteúdos às diversas áreas.
- C. as designações dos diferentes *layouts* que constituem as diferentes áreas da *CSS grid*.
- D. as designações dos diferentes *templates* que constituem as diferentes áreas da *CSS grid*.

☐

28. Considere o seguinte código HTML:

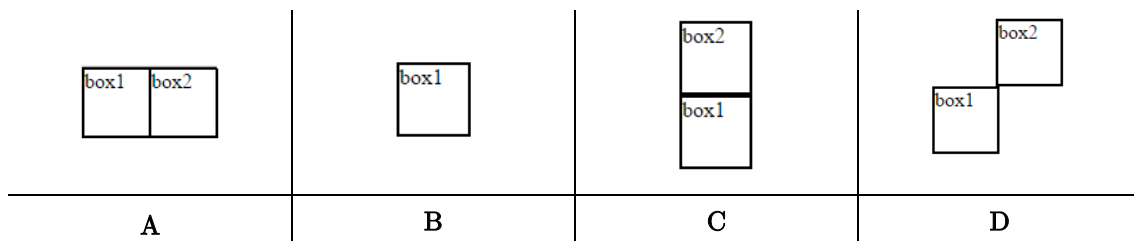
```
<div id="box1">box2</div>
<div id="box2">box1</div>
```

o código CSS:

```
div{
    border:2px black solid; background-color: white;
    width:50px;
    height:50px;}

```

assim como os possíveis outputs:



[0.4 Valores]

28.1. Identifique o output correto com base no seguinte código CSS.

```
#box1{position:static;left:50px;}
#box2{position:static;}

```

☐

[0.4 Valores]

28.2. Identifique o output correto com base no seguinte código CSS.

```
#box1{position:static;left:50px;}
#box2{position:relative;}

```

☐

[0.4 Valores]

28.3. Identifique o output correto com base no seguinte código CSS.

```
#box1{position:relative;}  
#box2{position:relative;}
```

☐

[0.4 Valores]

28.4. Identifique o output correto com base no seguinte código CSS.

```
#box1{position:absolute;}  
#box2{position:relative}
```

☐

[0.4 Valores]

29. A definição de estilos no interior dos *CSS media queries*:

- A. deve assegurar uma maior especificidade dos seletores de forma a conseguir alterar a formatação anteriormente aplicada a um determinado elemento.
- B. prevalece sobre os estilos definidos fora da *media query*, como tal a especificidade dos seletores é irrelevante, conseguindo sempre alterar a formatação anteriormente aplicada.
- C. não prevalece sobre os estilos definidos fora da *media query*, como tal a especificidade dos seletores é irrelevante, nunca conseguindo alterar a formatação anteriormente aplicada.
- D. deve assegurar uma menor especificidade dos seletores de forma a conseguir alterar a formatação anteriormente aplicada a um determinado elemento.

☐

[0.4 Valores]

30. No contexto do Sass, a declaração de funções envolve:

- A. *function* / *return*.
- B. *@function* / *@return*.
- C. *@function* / *@exit*.
- D. *@function* / *@quit*.

☐

[0.4 Valores]

31. No contexto do Sass, a reutilização de código pode ser efetuada através de:

- A. *@mixin*
- B. *@loop*
- C. *@media*
- D. *@reuse*

☐

[0.4 Valores]

32. O *JavaScript* é uma linguagem *loosely typed* uma vez que:

- A. na declaração das variáveis (*keyword var*) não é definido o tipo de variável.
- B. apenas podem ser declaradas variáveis dos tipos: *numeric* e *string*.
- C. na declaração das variáveis (*keyword variable*) não é definido o tipo de variável.
- D. apenas podem ser declaradas variáveis dos tipos: *numeric*, *string* e *boolean*.

☐

[0.4 Valores]

33. Em *JavaScript* as variáveis podem ser enquadradas em dois grandes grupos:

- A. *Numeric/Strings*.
- B. *Numeric/Arrays*.
- C. *Objects/Reference Types*.
- D. *Primitive Types/Reference Types*.

☐

[0.4 Valores]

34. Em *JavaScript* a declaração de variáveis:

- A. não é permitida.
- B. faz-se exclusivamente através de ***var***.
- C. pode ser efetuada através de ***var*** ou ***let***.
- D. pode ser efetuada através ***var***, ***let*** ou ***this***.

☐

[0.4 Valores]

35. A criação de um objeto (instância) com base num *constructor* é efetuado com base na *keyword*:

- A. *create*.
- B. *instance*.
- C. *new*.
- D. *void*.

☐

[0.4 Valores]

36. Em *JavaScript* um *Array*:

- A. pode armazenar diferentes tipos de dados e permite indexação cujo valor se inicia em 0.
- B. pode armazenar diferentes tipos de dados e permite indexação cujo valor se inicia em 1.
- C. não pode armazenar diferentes tipos de dados e permite indexação cujo valor se inicia em 1.
- D. não pode armazenar diferentes tipos de dados e permite indexação cujo valor se inicia em 0.

☐

[0.4 Valores]

37. Se utilizado na definição de um método, ***this***:

- A. refere-se ao objeto que contém o método e é usado como forma de aumentar o acoplamento direto entre objeto e método.
- B. refere-se ao *Window Object*.
- C. refere-se ao objeto que contém o método e é usado como forma de reduzir o acoplamento direto entre objeto e método.
- D. refere-se ao *Style Object*.

☐

[0.4 Valores]

38. O método *document.querySelector()* permite aceder a um elemento HTML através de seletores:

- A. CSS e dispensa indexação.
- B. *JavaScript* e requer indexação.
- C. CSS e requer indexação.
- D. *JavaScript* e dispensa indexação.

☐

[0.4 Valores]

39. Os *HTML event handlers*:

- A. não permitem uma separação clara entre o HTML e o *JavaScript*, sendo a definição de todo o código efetuada diretamente no HTML.
- B. permitem uma separação clara entre o HTML e o *JavaScript*, sendo a definição de todo o *event handling* efetuada diretamente no HTML.
- C. permitem uma separação clara entre o HTML e o *JavaScript*, sendo a definição de todo o *event handling* efetuada no script.
- D. não permitem uma separação clara entre o HTML e o *JavaScript*.

☐