

Contrôle continu – Corrigé

Conception de sites web dynamiques

Durée : 35 minutes

1. Qu'est-ce qu'un site web statique ? Donnez une réponse brève.

Un site web statique est constitué de pages HTML et CSS déjà présentes sur le serveur web. Elles sont prêtes à être envoyées vers l'utilisateur sans aucune modification.

2. Qu'est-ce qu'un site web dynamique (côté serveur) ? Donnez une réponse brève.

Un site web dynamique génère des pages HTML et CSS à la demande. Ces pages n'existaient pas avant la connexion de l'utilisateur.

3. Donnez trois exemples de balises orphelines.

```
<br>
<input>
<img>
```

4. Soit la ligne de code suivante : `Cliquez ici`

Donnez le nom de l'élément : a

Donnez le contenu de l'élément : Cliquez ici

Donnez le nom de l'attribut : href

Donnez la valeur de l'attribut : fichier.png

5. À l'aide des éléments *dl*, *dt* et *dd*, donnez le code HTML qui donnera l'affichage ci-dessous. Ne donnez que le contenu de l'élément *body*.

HTML

Langage de balisage utilisé pour créer les documents hypertextes et hypermédias du Web.

```
<dl>
  <dt>HTML</dt>
  <dd>
    Langage de balisage utilisé pour créer les documents
    hypertextes et hypermédias du Web.
  </dd>
</dl>
```

6. Le langage CSS sert-il à définir le fond ou la forme d'une page web ?

Le langage CSS sert à définir la forme d'une page web.

7. Donnez la ligne de code HTML qui permet d'afficher un lien (*http://www.google.fr*) sous forme d'image (*image.png*).

```
<a href="http://www.google.fr"></a>
```

8. Dans un formulaire, comment fait-on pour lier un libellé à une zone de saisie ?

L'attribut *for* du libellé doit être identique à l'identifiant (attribut *id*) de la zone de saisie.

9. Donnez le bout de code HTML (uniquement le code du libellé et celui de la liste déroulante) qui permet d'obtenir l'affichage ci-dessous. L'option *Mme* devra être sélectionnée par défaut.



```
<label for="civillite">Civilité :</label>
<select id="civillite" name="civillite">
  <option value="M">M.</option>
  <option value="Mme" selected>Mme</option>
  <option value="Mlle">Mlle</option>
</select>
```

Soit les trois modèles suivants :

Modèle 1	Modèle 2	Modèle 3
<pre>sélecteur { propriété_1: valeur_1; propriété_2: valeur_2; propriété_3: valeur_3; }</pre>	<pre>propriété { sélecteur_1: valeur_1; sélecteur_2: valeur_2; sélecteur_3: valeur_3; }</pre>	<pre>propriété { valeur_1: sélecteur_1; valeur_2: sélecteur_2; valeur_3: sélecteur_3; }</pre>

10. Dans un fichier CSS, quel modèle est valide parmi les trois ci-dessus ?

Le modèle 1 est valide.

11. Donnez le sélecteur qui sélectionne tous les éléments d'une classe *arbre* et un élément d'identifiant *feuille*.

.arbre, #feuille

12. Donnez le sélecteur qui sélectionne les éléments *p* situés après un élément *ul*.

ul ~ p

13. Donnez le sélecteur qui sélectionne les éléments *li* situés dans un élément de classe *rouge*.

.rouge li

14. Donnez le sélecteur qui sélectionne un élément *input* qui possède le *focus*.

input:focus

15. Donnez le sélecteur qui sélectionne les paragraphes situés immédiatement après un élément *h2* de classe *note*.

h2.note + p

16. Donnez le sélecteur qui sélectionne les éléments ayant un attribut *id*.

[id]

17. Donnez le sélecteur qui sélectionne les images possédant une infobulle.

img[title]

18. Donnez le sélecteur qui sélectionne un lien déjà visité et qui pointe vers l'URL « http://www.google.fr ».

a:visited[href="http://www.google.fr"]

19. Donnez le code HTML qui donnera l’affichage ci-dessous. Ne donnez que le contenu de l’élément *body*. Il n’est pas demandé d’afficher les bordures du tableau.

Légende

Titre 1	Titre 2	Titre 3
case 1.1	case 1.2	case 1.3
case 2.1 + 2.2		case 2.3
case 3.1	case 3.2	case 3.3

```
<table>
  <caption>Légende</caption>
  <tr>
    <th>Titre 1</th>
    <th>Titre 2</th>
    <th>Titre 3</th>
  </tr>
  <tr>
    <td>case 1.1</td>
    <td>case 1.2</td>
    <td>case 1.3</td>
  </tr>
  <tr>
    <td colspan="2">case 2.1 + 2.2</td>
    <td>case 2.3</td>
  </tr>
  <tr>
    <td>case 3.1</td>
    <td>case 3.2</td>
    <td>case 3.3</td>
  </tr>
</table>
```

20. Donnez le code HTML qui donnera l’affichage ci-dessous. Ne donnez que le contenu de l’élément *body*.

1. Premier élément de la liste.
2. Deuxième élément de la liste.
 - Sous-élément 1
 - Sous-élément 2
3. Troisième élément de la liste.

```
<ol>
  <li>Premier élément de la liste.</li>
  <li>
    Deuxième élément de la liste.
    <ul>
      <li>Sous-élément 1</li>
      <li>Sous-élément 2</li>
    </ul>
  </li>
  <li>Troisième élément de la liste.</li>
</ol>
```

21. Donnez le code CSS (sélecteurs + propriétés + valeurs) qui permet d’effectuer les opérations suivantes sur un paragraphe de classe *note* :

- Le texte doit s’afficher en italique.
- Le fond du texte doit s’afficher en vert avec une transparence de 50 %.
- Le texte doit être justifié.
- La taille du texte doit être de 25 pixels.

```
p.note
{
  font-style: italic;
  background-color: rgba(0, 255, 0, 0.5);
  text-align: justify;
  font-size: 25px;
}
```

22. Que fait la méthode *querySelector()* ? (Quels sont ses arguments et que renvoie-t-elle ?)

La méthode *querySelector()* prend comme argument un sélecteur CSS (sous forme de chaîne de caractères) et renvoie un objet contenant le premier élément de la page HTML qui correspond à ce sélecteur.

23. Que faut-il ajouter à la balise *<body>* pour lancer une fonction *init()* lorsque le corps de la page est complètement chargé ? Donnez uniquement une ligne de code HTML (balise ouvrante + attribut).

```
<body onLoad="init();">
```

24. En JavaScript, soit une variable *p* contenant un objet sur un paragraphe (élément *p*). Donnez une instruction JavaScript qui permet de mettre en gras le texte du paragraphe.

```
p.style.fontWeight = "bold";
```

25. Donnez la valeur des variables *a*, *b*, *c* et *d* après l'exécution du code JavaScript suivant.

```
var a = "6" + 2;  
var b = "6" - 2;  
var c = "6" * 2;  
var d = "6" / 2;
```

```
a = "62"  
b = 4  
c = 12  
d = 3
```

26. Donnez la valeur des variables *ta*, *tb*, *tc*, *td* et *te* après l'exécution du code JavaScript suivant.

```
var a = 5;          var ta = typeof a;  
var b = true;       var tb = typeof b;  
var c = "Hello";    var tc = typeof c;  
var d;              var td = typeof d;  
                   var te = typeof e;
```

```
ta = "number"  
tb = "boolean"  
tc = "string"  
td = "undefined"  
te = "undefined"
```

Soit le code HTML suivant :

```
<a href="#" onClick="ChangeColor();">Changer la couleur</a>
```

27. Donnez le code JavaScript de la fonction *ChangeColor()* qui colore en rouge le texte de tous les paragraphes de la classe *important*. Vous utiliserez la méthode *querySelectorAll()* de l'objet *document*.

```
function ChangeColor()
{
    var ps = document.querySelectorAll("p.important");
    for (var i = 0; i < ps.length; i++)
        ps[i].style.color = "red";
}
```

28. La fonction *ChangeColor()* est-elle exécutée côté serveur où côté client ?

La fonction *ChangeColor()* est exécutée côté client.