

Examen (Session 1) – Corrigé

Conception de sites web dynamiques

Durée : 1 h 30

1. Si un serveur contient uniquement des pages avec l'extension PHP. Le client qui se connecte à ce serveur recevra-t-il des pages au format PHP ?

Non. Les scripts PHP seront exécutés par le serveur qui enverra au client uniquement le résultat de ces scripts (généralement des pages au format HTML).

2. Le langage HTML est-il un langage compilé ? Si oui, quel logiciel compile ce langage ?

Le langage HTML n'est pas un langage compilé. Il s'agit d'un langage interprété.

3. La première ligne d'une page HTML 5 est-elle une balise HTML ? Si oui, laquelle ?

Non, il s'agit d'un *Doctype (Document Type)* qui indique au navigateur qu'il s'agit d'un document au format HTML 5.

4. Quel est l'élément parent de l'élément *title*.

L'élément parent de l'élément *title* est l'élément *head*.

5. La finalité de l'élément *em* est-elle de mettre le texte en italique ou de le mettre en gras, ou bien ni l'un ni l'autre (c'est-à-dire ni en italique, ni en gras) ?

La finalité de l'élément *em* est de préciser une emphase sémantique. Ce n'est donc ni de mettre le texte en italique (comportement par défaut), ni de mettre le texte en gras.

6. Qu'afficheront dans le navigateur les lignes de code HTML suivantes ?

```
<body>
  <p>Syntaxe de l'élément : &lt;em>&lt;/em></p>
</body>
```

Syntaxe de l'élément : ``

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

- Vente :
 1. Lit
 2. Armoire
- Achat (à voir plus tard)

```
<ul>  
  <li>  
    Vente :  
    <ol>  
      <li>Lit</li>  
      <li>Armoire</li>  
    </ol>  
  </li>  
  <li>Achat (à voir plus tard)</li>  
</ul>
```

8. Donnez une ligne de code HTML qui crée un lien vers un élément possédant l’identifiant *part_1*. L’élément à atteindre se trouve dans la même page que le lien. Le texte du lien sera *Partie 1*.

```
<a href="#part_1">Partie 1</a>
```

9. À quoi servent les attributs *alt* et *title* de l’élément *img* ? Donnez une réponse brève.

L’attribut *alt* sert à afficher un texte alternatif au cas où l’image ne serait pas affichée.

L’attribut *title* sert à afficher une infobulle lors du survol de l’image par le pointeur de la souris.

10. L’attribut *alt* de l’élément *img* est-il obligatoire ?

Oui, l’attribut *alt* de l’élément *img* est obligatoire.

Soit les deux lignes suivantes :

```
<p><b>Hello World</b></p>
<b><p>Hello World</p></b>
```

11. Quelle(s) ligne(s) est (sont) valide(s) ? (Aucune, la première, la seconde, les deux.)

La première ligne est valide. Dans la seconde ligne, un élément de type *inline* (*b*) ne peut pas contenir un élément de type *block* (*p*).

12. Donnez le code CSS (sélecteurs + propriétés + valeurs) qui permet d'effectuer les opérations suivantes sur les paragraphes.

- La taille du texte doit être de 15 pixels.
- Le fond du texte doit s'afficher en rouge.
- Une image doit s'afficher en arrière-plan. Le fichier contenant l'image est *figure.png* et est situé dans le même répertoire que le fichier CSS.
- L'image de fond doit se répéter uniquement verticalement.

```
p
{
    font-size: 15px;
    background-color: rgb(255, 0, 0);
    background-image: url("figure.png");
    background-repeat: repeat-y;
}
```

13. Donnez les deux balises universelles (sans effet par défaut).

Les deux balises universelles sont `<div>` et ``.

14. Quelle balise universelle est de type *block* ?

La balise universelle de type *block* est `<div>`.

15. Donnez deux mots clés différents en CSS pour gérer l'opacité.

rgba
opacity

16. 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.

17. 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.

18. 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";`

Soit le code HTML suivant :

`Changer la couleur`

19. 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";
}
```

20. En JavaScript, donnez le nom des méthodes qui permettent de réaliser les actions suivantes :

Créer un élément HTML : *createElement()*

Créer un nœud textuel : *createTextNode()*

Ajouter un nœud fils à un nœud : *appendChild()*

Supprimer un nœud fils : *removeChild()*

21. Donnez le sélecteur qui sélectionne les éléments d'une classe *animal* et l'élément d'identifiant *ours*.

`.animal, #ours`

22. Donnez le sélecteur qui sélectionne les liens de classe *out*.

`a.out`

23. Donnez le sélecteur qui sélectionne les listes ordonnées situées après une image.

`img ~ ol`

24. Donnez le sélecteur qui sélectionne les éléments *strong* situés dans une liste non ordonnée de classe *cuisine*.

`ul.cuisine strong`

25. Donnez le sélecteur qui sélectionne les éléments ayant un attribut *src*.

`[src]`

26. Donnez le sélecteur qui sélectionne les hyperliens déjà visités situés immédiatement après une image.

`img + a:visited`

27. Donnez le sélecteur qui sélectionne les éléments *em* qui sont directement dans un paragraphe.

`p > em`

28. Donnez le sélecteur qui sélectionne les éléments *input* de nom *user* situés dans un paragraphe.

`P input[name="user"]`

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

`img[title]`

30. Donnez le sélecteur qui sélectionne les hyperliens qui n'ont jamais été visités et les éléments *input* de type *text*.

`a:link, input[type="text"]`

31. Soit le code PHP ci-dessous. Inscrivez ce qu'affichent les trois instructions `echo`.

```
$foo = "0";           echo gettype($foo);  
$foo += 2;           echo gettype($foo);  
$foo = $foo + 1.3;    echo gettype($foo);
```

Instruction `echo 1` : `string`

Instruction `echo 2` : `integer`

Instruction `echo 3` : `double`

32. Si une variable PHP est non déclarée ou sans valeur, alors elle est considérée comme étant vide. Donnez trois autres cas possibles où une variable PHP est considérée comme étant vide.

```
$var1 = 0;  
$var2 = "";  
$var3 = false;
```

33. Si un utilisateur tape `http://www.parisdescartes.com/index.php` dans la barre d'adresse de son navigateur et affiche le code source de la page. Le code source qui s'affiche contient-il du code HTML, du code PHP ou les deux ?

Le code qui s'affiche contient du code HTML et pas de code PHP.

34. Donnez une instruction PHP qui définit une constante *UNIV* avec la chaîne de caractères « Paris 5 ».

```
define("UNIV", "Paris 5");
```

35. En supposant qu'une constante *UNIV* contient la chaîne de caractères « Paris 5 », pourquoi l'instruction PHP suivante ne fonctionne-t-elle pas ? Que faut-il modifier ?

```
echo "Contenu de la constante : ".$UNIV;
```

L'accès au contenu d'une constante se fait sans le caractère « \$ » :
`echo "Contenu de la constante :".UNIV;`

36. Soit le fichier *test.php* ci-dessous :

```
<?php
echo "Bonjour " . $_GET["name"];
echo "Erreur numéro " . $_GET["error"];
```

En supposant que le nom d'hôte est *parisdescartes.fr* et que le fichier est situé dans le répertoire racine du serveur, donnez l'URL qui permet d'obtenir l'affichage suivant :

```
Bonjour David
Erreur numéro 81
```

```
http://www.parisdescartes.fr/test.php?name=David&error=81
```

37. Où se trouve l'interpréteur PHP ?

L'interpréteur PHP se trouve sur le serveur.

38. La différence entre les opérateurs `==` et `===` du langage PHP est-elle identique à celle de ces mêmes opérateurs en langage JavaScript ? Autrement dit, ces opérateurs fonctionnent-ils de la même façon en PHP et en JavaScript ?

Oui, la différence est la même. Ces opérateurs fonctionnent de la même façon en PHP et en JavaScript.

39. Soit le code HTML ci-dessous. Écrivez le code PHP qui affiche « Oui » si la case 1 est cochée ou « Non » si la case 1 n'est pas cochée. On suppose que le formulaire a été soumis correctement.

```
<form method="post" action="traitement.php">
  <input type="checkbox" name="Case_1"> Oui <br>
  <input type="submit" value="Valider">
</form>
```

```
if (isset($_POST['Case_1']))
    echo "Oui";
else
    echo "Non";
```

40. Soit le code HTML ci-dessous. Écrivez le code PHP qui affiche la valeur du bouton radio qui a été coché. Par exemple, si le bouton 2 est coché, il faudra afficher « choix_2 ». Si aucun bouton n'est coché, rien ne devra s'afficher. On suppose que le formulaire a été soumis correctement.

```
<form method="post" action="traitement.php">
  <input type="radio" name="choix" value="choix_1"> Choix 1 <br>
  <input type="radio" name="choix" value="choix_2"> Choix 2 <br>
  <input type="radio" name="choix" value="choix_3"> Choix 3 <br>
  <input type="submit" value="Valider">
</form>
```

```
echo $_POST['choix']
```

41. Donnez l'instruction PHP qui permet de sauvegarder le cookie suivant :

- Nom du cookie : pseudo
- Valeur du cookie : cobra
- Durée de vie du cookie : 1 heure

```
setcookie('pseudo', 'cobra', time() + 3600);
```

42. Quelle est la condition à respecter, afin que la fonction *setcookie()* fonctionne correctement ?

L'appel à la fonction *setcookie()* doit se faire avant tout affichage HTML.

43. Soit la fonction PHP *explode(\$a, \$b)* qui renvoie un tableau de chaînes. Chaque chaîne du tableau est une sous chaîne de la chaîne *b*. Les sous-chaînes sont extraites en utilisant le séparateur *a*. Que renvoie la fonction PHP ci-dessous pour l'appel suivant (respectez la casse) :

```
f("Ceci est un exemple.");
```

```
function f($s)
{
    $t = explode(" ", $s);

    foreach ($t as $w)
        $a .= strtoupper($w[0]);

    return $a;
}
```

Cette fonction renvoie les premières lettres (converties en majuscule) de chaque mot de la chaîne passée en paramètre.

L'appel `f("Ceci est un exemple.")` renvoie la chaîne de caractères « CEUE ».

44. Quel type de requête est-il préférable de faire afin d'éviter les injections SQL ?

Il est préférable d'effectuer des requêtes préparées.

45. Afin d'accéder à une base de données MySQL, est-il plutôt recommandé d'utiliser les instructions PHP préfixées par *mysql* ou celles préfixées par *mysqli* ?

Il est plutôt recommandé d'utiliser les instructions PHP préfixées par *mysqli*.

46. Donnez le type et la valeur de la variable *\$foo* après l'instruction suivante :

```
$foo = 5 + "10 Little Piggies";
```

Type : *Integer*

Valeur : 15