

Examen (Session 1) – Corrigé

Conception de sites web dynamiques

Durée : 1 h 30

Nom : Prénom : No étudiant :

Cadre réservé au correcteur.

Ne pas justifier les réponses sauf si cela est explicitement demandé.

1. Un site contenant 100 fichiers PHP et un seul fichier HTML est-il statique ou dynamique ?

Le site est dynamique (un seul fichier PHP suffit à rendre un site dynamique).

2. Un site contenant 100 fichiers HTML et un seul fichier PHP est-il statique ou dynamique ?

Le site est dynamique (un seul fichier PHP suffit à rendre un site dynamique).

3. Quelle balise universelle est de type *inline* ?

La balise universelle de type *inline* est ``.

4. Quel est l'élément parent de l'élément *body* ?

L'élément parent de l'élément *body* est l'élément *html*.

5. Quelle est la première balise d'une page HTML ?

La première balise d'une page HTML est la balise `<html>`.

6. Quelles balises permettent de remplacer explicitement du texte ? Par exemple :

Mon alcool préféré est le `<?>whisky</?>` `<?>rhum</?>`.

Il s'agit des balises `` et `<ins></ins>`.

Mon alcool préféré est le `whisky` `<ins>rhum</ins>`.

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

- Fruits :
 - Orange
 - Pomme
- Légumes :
 1. Carotte
 2. Poireau

```
<ul>
  <li>Fruits :
    <ul>
      <li>Orange</li>
      <li>Pomme</li>
    </ul>
  </li>
  <li>Légumes :
    <ol>
      <li>Carotte</li>
      <li>Poireau</li>
    </ol>
  </li>
</ul>
```

8. Donnez le code HTML qui permet de créer un lien vers une adresse e-mail. L’adresse e-mail est « contact@paris5.fr » et le texte à afficher est « Nous contacter ».

```
<a href="mailto:contact@paris5.fr">Nous contacter</a>
```

9. L’attribut *title* de l’élément *img* est-il obligatoire ?

Non, l’attribut *title* de l’élément *img* n’est pas obligatoire.

10. 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.

11	12	13
21	22	
31	32	33
41	42	

```
<table>
  <tr>
    <th>11</th>
    <th>12</th>
    <th>13</th>
  </tr>
  <tr>
    <td>21</td>
    <td colspan="2">22</td>
  </tr>
  <tr>
    <td>31</td>
    <td>32</td>
    <td rowspan="2">33</td>
  </tr>
  <tr>
    <td>41</td>
    <td>42</td>
  </tr>
</table>
```

11. Donnez le bout de code HTML qui permet d'obtenir l'affichage ci-dessous. Les libellés « Oui » et « Non » seront liés aux boutons. Ne donnez pas le code de tout le formulaire (uniquement de ce qui s'affiche).

Etes vous d'accord ?

- ☐ Oui
☐ Non

```
<label>Etes vous d'accord ?</label>  
<br>  
<input type="radio" id="oui" name="accord" value="oui">  
<label for="oui">Oui</label>  
<br>  
<input type="radio" id="non" name="accord" value="non">  
<label for="non">Non</label>
```

12. Expliquez brièvement ce que sont les positions *relative*, *absolute* et *fixed*.

Relative : Placement dans le flux normal puis décalage.

Absolute : Placement en dehors du flux normal en référence à une boîte conteneur.

Fixed : Placement en dehors du flux normal en référence à la fenêtre du navigateur.

13. Soit la ligne de code CSS suivante :

```
color: rgb(0, 0, 255);
```

Donnez trois syntaxes différentes possibles afin d'effectuer exactement la même chose.

```
color: blue;  
color: rgb(0%, 0%, 100%);  
color: #0000ff;  
color: #00f;  
color: rgba(0, 0, 255, 1);
```

14. Donnez le code HTML qui permet d'afficher un lecteur vidéo avec une barre de contrôle. La résolution de la vidéo est de 320x240 pixels. Deux fichiers vidéo se trouvent dans le même répertoire que le fichier HTML : « movie.mp4 » et « movie.ogg ». On suppose que le navigateur est récent et qu'il permet l'affichage de la vidéo.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

15. Donnez le code CSS (sélecteurs + propriétés + valeurs) qui permet d'effectuer les opérations suivantes sur un élément d'identifiant *note* :

- La famille de la police doit être *verdana*.
- Le texte doit être souligné.
- La taille de la police doit être de 15 pixels.
- La police doit être en gras.

```
#note
{
  font-family: verdana;
  text-decoration: underline;
  font-size: 15px;
  font-weight: bold;
}
```

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

- Le texte doit s'afficher en rouge.
- Le fond du texte doit s'afficher en bleu avec une transparence de 70 %.
- Le texte doit être justifié.
- Le texte doit s'afficher avec un décalage de 10 pixels par rapport à sa position initiale (le décalage pourra se faire soit vers la gauche, soit vers la droite, le choix vous appartient).

```
p
{
  color: red;
  background-color: rgba(0, 0, 255, 0.7);
  text-align: justify;
  position: relative;
  left: 10px;
}
```

17. Donnez le sélecteur qui sélectionne les éléments d'une classe *number* et l'élément d'identifiant *star*.

`.number, #star`

18. Donnez le sélecteur qui sélectionne les liens d'une classe *number* situé dans un élément d'identifiant *star*.

`#star a.number`

19. Donnez le sélecteur qui sélectionne les listes ordonnées situées après une image de classe *sky*.

`img.sky ~ ol`

20. Donnez le sélecteur qui sélectionne les éléments *strong* situés dans une liste non ordonnée de classe *animal* et un élément d'identifiant *horse*.

`ul.animal strong, #horse`

21. Donnez le sélecteur qui sélectionne les éléments ayant un attribut *href*.

`[href]`

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

`img + a:visited`

23. Donnez le sélecteur qui sélectionne les éléments *strong* de classe *horse* qui sont directement dans un paragraphe.

`p > strong.horse`

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

`P input[name="user"]`

25. Donnez le sélecteur qui sélectionne les liens possédant une infobulle.

`a[title]`

26. Donnez le sélecteur qui sélectionne les hyperliens qui n'ont jamais été visités et les zones de saisie de mot de passe.

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

27. 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()*

28. Soit le code suivant :

```
<body>
  <p id="letexte">
    Texte avec <b>des caractères gras</b>
    et <u>du texte souligné</u></p>
  <script>
    var p = document.getElementById("letexte");
    alert(p.childNodes.length);
    alert(p.firstChild.nodeValue);
    alert(p.childNodes[1].childNodes[0].nodeValue);
  </script>
</body>
```

Qu'afficheront les appels à la fonction *alert()* (inutile de préciser les espaces superflus) ?

1^{er} appel : 4

2^e appel : Texte avec

3^e appel : des caractères gras

29. Donnez quelques instructions en JavaScript qui permettent de fixer la taille de la police de tous les paragraphes à 30 pixels.

```
var elements = document.querySelectorAll("p");
for (var i = 0; i < elements.length; i++)
  elements[i].style.fontSize = "30px";
```

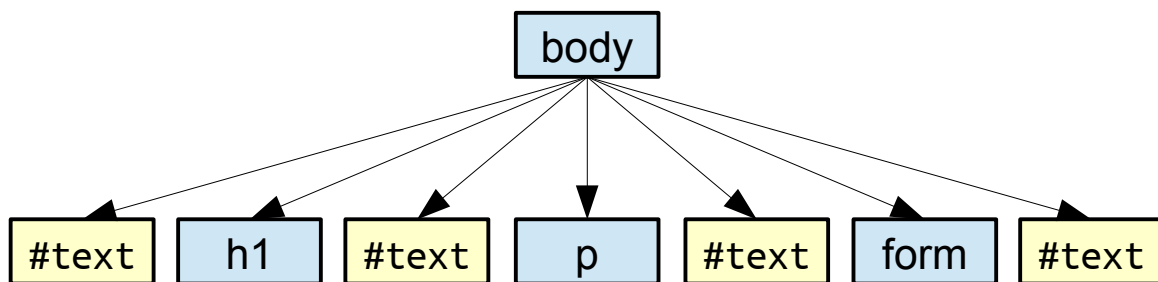
30. Donnez quelques instructions en JavaScript qui permettent d'ajouter une bordure de 1 pixel sur tous les liens.

```
var elements = document.querySelectorAll("a");
for (var i = 0; i < elements.length; i++)
  elements[i].style.border = "solid 1px";
```

Soit le code suivant :

```
<html>
  <head>
    <title>Formulaire</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Formulaire</h1>
    <p>Cochez si vous être d'accord</p>
    <form name="form" method="post">
      <input type="checkbox" id="ok">
      <label for="OK">OK</label>
    </form>
  </body>
</html>
```

31. À partir du code source ci-dessus, donnez la représentation graphique du DOM sous la forme d'un arbre en prenant l'élément *body* comme nœud racine (ne représentez pas les nœuds parents de l'élément *body*). Représentez uniquement l'élément *body* et ses nœuds enfants (les enfants des enfants ne devront pas être représentés). **Représentez les nœuds d'éléments ainsi que les nœuds textuels.**



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

```
var a = "8" - 2;
var b = "8" + 2;
var c = "8" * 2;
var d = "8" / 2;
```

```
a = 6
b = "82"
c = 16
d = 4
```

33. Donnez le code HTML qui permet d'exécuter la fonction *traitement()* lors d'un survol du pointeur de la souris sur un paragraphe qui contient uniquement le texte « Hello World ». Une ligne de code est suffisante (balise ouvrante + attribut + contenu + balise fermante).

```
<p onmouseover="traitement();">Hello World</p>
```


34. Donnez les instructions PHP qui affichent la valeur des variables *nom* et *age* à partir de l'URL ci-dessous. L'affichage se fera à l'aide de l'instruction *echo*.

```
http://www.parisdescartes.fr?nom=roger&age=25
```

```
echo "nom : " . $_GET['nom'];  
echo "<br />  
echo "age : " . $_GET['age'];
```

35. Inscrivez ce qu'affichent les trois instructions PHP suivantes :

```
$x = 10;  
echo "La variable \$x vaut $x";           // Instruction 1  
echo "La variable ".$x." vaut $x";       // Instruction 2  
echo 'La variable $x vaut '.$x;          // Instruction 3
```

Instruction 1 : La variable \$x vaut 10

Instruction 2 : La variable 10 vaut 10

Instruction 3 : La variable \$x vaut 10

36. Donnez une instruction PHP qui définit une constante *ORDER* avec la chaîne de caractères « Go ».

```
define("ORDER", "Go");
```

37. Quelle est la différence entre les fonctions *pg_fetch_row()* et *pg_fetch_assoc()* ?

pg_fetch_row() renvoie une ligne courante d'un jeu de résultats d'une requête PostgreSQL sous la forme d'un tableau indicé.

pg_fetch_assoc() fait la même chose, mais renvoie la ligne courante sous la forme d'un tableau associatif.

38. Donnez au moins deux constantes magiques en langage PHP.

```
__LINE__ : La ligne courante dans le fichier.  
__FUNCTION__ : Le nom de la fonction.
```

39. Où se trouve l'interpréteur PHP ?

L'interpréteur PHP se trouve sur le serveur.

40. Qu'affiche le code ci-dessous ?

```
class A {public $x = "Oui";}
$a = new A();
$b = $a;
$c = clone $a;
$c->x = "Non"; echo $a->x; // Instruction 1
$b->x = "Si"; echo $a->x; // Instruction 2
```

Inscrivez uniquement ce qu'affichent les instructions (et rien d'autre) :

Instruction 1 : Oui

Instruction 2 : Si

41. Soit le code ci-dessous :

```
<form method="get" action="traitement.php">
Prénom : <input type="text" name="prenom" id="prenom"><br>
Nom : <input type="text" name="nom" id="nom"><br>
<input type="submit" value="Soumettre">
</form>
```

Donnez le code PHP qui permet d'afficher le nom et le prénom qui ont été saisis par l'utilisateur. Si l'utilisateur n'a rien saisi, le message « Aucune information n'a été saisie. » sera affiché. On suppose que l'utilisateur n'a rien saisi si les variables n'ont pas été transmises au script PHP. L'affichage se fera à l'aide de l'instruction *echo*.

```
if (isset($_GET['prenom']) && isset($_GET['nom']))
    echo $_GET['prenom']. " ".$_GET['nom'];
else
    echo "Aucune information n'a été saisie.";
```

42. 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);
```

43. Donnez le code PHP qui affichera puis supprimera sans délai le cookie sauvegardé dans la question précédente. L'affichage se fera à l'aide de l'instruction *echo*.

```
echo $_COOKIE['pseudo']
setcookie('pseudo', '', time() - 1)
```

44. Soit les informations de connexion et d'authentification suivantes :

- Nom de l'hôte : **localhost**
- Nom de la base de données : **database**
- Nom d'utilisateur : **gdupont**
- Mot de passe : **uiot5434**

Sachant que le constructeur d'un objet PDO prend les arguments suivants :

PDO::__construct('<prefix>:host=<host>;dbname=<dbname>', '<login>', '<password>');

Donnez le code PHP qui permet de se connecter à une base de données PostgreSQL à l'aide d'un objet PDO. Si une erreur survient lors de la connexion, le script PHP devra s'arrêter en affichant un message d'erreur.

```
try
{
    $pgdB = new PDO('pgsql:host=localhost;dbname=database', 'gdupont', 'uiot5434');
}
catch (Exception $e)
{
    exit($e->getMessage());
}
```

45. Soit la requête préparée suivante :

```
$query = "SELECT * FROM customer WHERE name = ? AND age < ?";
```

Sachant que la connexion à la base de données a déjà été faite à l'aide d'un objet PDO contenu dans une variable *\$pdo*, donnez le code PHP qui permet de récupérer le résultat de cette requête dans une variable *\$result* avec comme nom « roger » et comme âge « 30 ».

```
$result = $pdo->prepare($query);
$params = array("roger", 30);
$result->execute($params);
```

46. En PHP, à quoi sert la fonction *bindParam()* ? Donnez une réponse brève.

La fonction *bindParam()* sert à lier des variables PHP aux différents paramètres d'une requête préparée.

47. Soit le fichier *count.php* qui incrémente le contenu d'une variable de session appelée *count*. Rédigez **l'intégralité** du fichier *count.php*. Si la variable de session n'existe pas, elle devra être créée et initialisée à la valeur 1. Si elle existe, elle devra être incrémentée de 1. Ce fichier ne contiendra que des instructions PHP (pas de HTML). Vous placerez également la pseudo balise d'ouverture d'un code PHP en début de fichier.

```
<?php
session_start();

if (isset($_SESSION['count']) === false)
    $_SESSION['count'] = 1;
else
    $_SESSION['count']++;
```

48. Expliquez très brièvement l'intérêt principal de l'extension PDO (*PHP Data Objects*).

PDO fournit une interface d'abstraction à l'accès de données, ce qui signifie que l'on utilise les mêmes fonctions pour exécuter des requêtes ou récupérer les données, quelle que soit la base de données utilisée.

49. Que signifient les initiales MVC dans « Architecture MVC » ?

MVC est l'abréviation de « Modèle Vue Contrôleur ».