

Technologies Web 2 - YSINL2C1

Contrôle terminal - Session 1

Durée : 1h

Ni documents, ni calculatrices, ni ordinateurs autorisés

Mai 2023

Consignes

L'épreuve comporte 2 parties, chacune notée sur 10.

Pour la partie 1 (QCM), indiquer pour chaque question votre réponse dans le cadre ci-dessous.

Détacher la page 1 que vous rendrez avec votre copie.

Seule cette page sera prise en compte.

Indiquer votre n° de table sur cette page.

La partie 2 doit être rédigée sur une copie anonyme.

N° table : _____

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	

PARTIE 1 (10 points)

Cette partie comporte 10 questions, numérotées de 1 à 10. Chaque question comporte quatre réponses possibles, étiquetées de A à D. **Chaque question a une et une seule bonne réponse.** Vous devez choisir une seule réponse pour chaque question et reporter votre choix dans le tableau page 1.

Chaque bonne réponse rapporte 1 point, chaque mauvaise réponse, réponse multiple ou absence de réponse enlève 0.25 point.

Tous les script sont supposés être en mode strict (c'est à dire commencer par "use strict").

Certaines questions réfèrent au code HTML ci-dessous. On suppose que la fonction `init()` est présente dans le fichier `examen.js` inclut par la balise `<script>` de l'en-tête de la page.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Examen TW2 - QCM</title>
  <script src="examen.js"></script>
</head>
<body>
  <header>
    <h1>Examen Javascript</h1>
    <p>L'examen dure 1h</p>
  </header>
  <main>
    <nav id="liens">
      <h2>Liens utiles</h2>
      <ul>
        <li><a href="https://ensweb.users.info.unicaen.fr"
id="ensweb">Pages du cours</a></li>
        <li class="ext"><a href="https://google.com">Google</a></li>
        <li class="ext"><a href="https://caniuse.com">Canibase</a></li>
      </ul>
    </nav>
    <article>
      <h2 id="titre-article">Un article</h2>
      <p>Le texte de l'article qui discute de :</p>
      <ul>
        <li>ceci</li>
        <li>et cela</li>
      </ul>
    </article>
  </main>

  <script>
    "use strict";
    init();
  </script>
</body>
</html>
```

Question 1

Que va afficher la console Javascript du navigateur lors de l'exécution du code suivant :

```
let nom = "Lachenal";  
if (nom.length > 8) {  
  let message = "C'est un nom long";  
} else {  
  let message = "C'est un petit nom";  
}  
console.log(message);
```

Réponse A

C'est un nom long

Réponse B

C'est un petit nom

Réponse C

Uncaught ReferenceError: message is not defined

Réponse D

Uncaught TypeError: nom.length is not a function

Question 2

On écrit une fonction qui fait la somme de 2 nombres et on l'utilise pour afficher en console la somme de 3 et 9.

Parmi les 4 possibilités ci-dessous, quel code est le seul qui est acceptable comme correct ?

Plusieurs réponses fonctionnent mais une seule obéit aux bonnes pratiques de programmation vues au cours du semestre.

Réponse A

```
let a = 3;
let b = 9;
function somme() {
  let c = a + b;
}
somme();
console.log(c);
```

Réponse B

```
let a = 3;
let b = 9;
function somme() {
  return a + b;
}
console.log(somme(3, 9));
```

Réponse C

```
function somme(a, b) {
  return a + b;
}
console.log(somme(3, 9));
```

Réponse D

```
function somme(a, b) {
  let c = a + b;
  console.log(c);
}
somme(3, 9);
```

Question 3

La variable `langages` contient une liste d'objets décrivant des langages de programmation, avec 2 propriétés `nom` et `type` (interprété ou compilé) :

```
let langages = [  
  { "nom": "javascript", "type": "interprété" },  
  { "nom": "java", "type": "compilé" },  
  { "nom": "php", "type": "interprété" },  
  { "nom": "C", "type": "compilé" },  
  ...  
];
```

On veut écrire une fonction `filtrer` qui prend en arguments une liste de langages et un type et qui renvoie les noms des langages du type demandé.

Quel code faut-il écrire pour que la commande `console.log(filtrer(langages, "compilé"))` produise par exemple l'affichage : `Array ["java", "C"]`

Réponse A

```
function filtrer(liste, type) {  
  let res = [];  
  for (let lang in liste) {  
    if (lang['type'] == type) {  
      res.push(lang);  
    }  
  }  
  return res;  
}
```

Réponse B

```
function filtrer(liste, type) {  
  let res = [];  
  for (let i = 0; i < liste.length; i++) {  
    if (liste[i]['type'] == type) {  
      res.push(liste[i]);  
    }  
  }  
  return res;  
}
```

Réponse C

```
function filtrer(liste, type) {  
  let res = [];  
  for (let lang in liste) {  
    if (lang.type == type) {  
      res.push(lang.nom);  
    }  
  }  
  return res;  
}
```

Réponse D

```
function filtrer(liste, type) {  
  let res = [];  
  for (let i = 0; i < liste.length; i++) {  
    if (liste[i]['type'] == type) {  
      res.push(liste[i]['nom']);  
    }  
  }  
  return res;  
}
```

Question 4

En utilisant la page web donnée en introduction, déterminer le code placé dans la fonction `init()` qui permet de récupérer les éléments `` de l'article et de les afficher en console.

Réponse A

```
let lesLis = document.querySelectorAll("article li");  
console.log(lesLis);
```

Réponse B

```
let lesLis = document.querySelectorAll("#article li");  
console.log(lesLis);
```

Réponse C

```
let lesLis = document.querySelector("article li");  
console.log(lesLis);
```

Réponse D

```
let lesLis = document.getElementById("article").children("li");  
console.log(lesLis);
```

Question 5

On veut afficher en console les URLs des liens de navigation qui sont dans un de classe ext. Quel code placé dans la fonction init() fait cela ?

Réponse A

```
let liens = document.querySelectorAll(".ext a");
for (let i = 0; i < liens.length; i++) {
  console.log(liens[i].getAttribute('href'));
}
```

Réponse B

```
let liens = document.querySelectorAll("#liens a");
for (let i = 0; i < liens.length; i++) {
  console.log(liens[i].getAttribute('href'));
}
```

Réponse C

```
let liens = document.querySelectorAll("#liens a");
console.log(liens.href);
```

Réponse D

```
let liens = document.querySelectorAll(".ext");
for (let i = 0; i < liens.length; i++) {
  console.log(liens[i].href);
}
```

Question 6

On utilise le code HTML de la page web donnée en introduction. Quel Javascript faut-il écrire dans la fonction `init()` pour ajouter en fin de page un élément `<footer>` avec un texte ?

Réponse A

```
let footer = createElement('footer');  
footer.textContent = "Le pied de page";  
body.appendChild(footer);
```

Réponse B

```
let footer = document.createElement('footer');  
footer.textContent = "Le pied de page";  
document.appendChild(footer);
```

Réponse C

```
let footer = document.createElement('footer');  
footer.textContent = "Le pied de page";  
document.body.appendChild(footer);
```

Réponse D

```
let footer = createElement('footer');  
footer.textContent = "Le pied de page";  
body.addChild(footer);
```

Question 7

On utilise le code HTML de la page web donnée en introduction. Quel code Javascript faut-il écrire dans la fonction `colorer` et dans la fonction `init()` pour que chaque titre h2 capte le clic et devienne alors rouge sur fond noir ?

Réponse A

```
function init() {
  let titres = document.querySelectorAll('h2');
  for (let i = 0; i < titres.length; i++) {
    titres[i].addEventListener('click', colorer)
  }
}

function colorer(event) {
  let titre = event.currentTarget;
  titre.style.color = 'red';
  titre.style.backgroundColor = 'black';
}
```

Réponse B

```
function init() {
  let titres = document.querySelectorAll('h2');
  for (let i = 0; i < titres.length; i++) {
    titres[i].addEventListener('click', colorer)
  }
}

function colorer(event) {
  titre[i].color = 'red';
  titre[i].style.backgroundColor = 'black';
}
```

Réponse C

```
function init() {
  let titres = document.querySelectorAll('h2');
  titres.addEventListener('click', colorer)
}

function colorer(event) {
  let titre = event.currentTarget;
  titre.style.color = 'red';
  titre.style.backgroundColor = 'black';
}
```

Réponse D

```
function init() {  
  let titres = document.querySelectorAll('h2');  
  titres.addEventListener('click', colorer())  
}  
function colorer(event) {  
  let titre = event.currentTarget;  
  titre.style.color = 'red';  
  titre.style.backgroundColor = 'black';  
}
```

Question 8

Si le code Javascript suivant est écrit dans la fonction `init()`, quel sera le HTML qui en résulte dans l'inspecteur du navigateur ?

```
let articleLis = document.querySelectorAll("article li");
for (let i = 0; i < articleLis.length; i++) {
  articleLis[i].dataset.numeroLi = i;
}
```

Réponse A

```
<ul>
  <li numeroLi="0">ceci</li>
  <li numeroLi="1">et cela</li>
</ul>
```

Réponse B

```
<ul>
  <li data-numeroLi="0">ceci</li>
  <li data-numeroLi="1">et cela</li>
</ul>
```

Réponse C

```
<ul>
  <li numero-li="0">ceci</li>
  <li numero-li="1">et cela</li>
</ul>
```

Réponse D

```
<ul>
  <li data-numero-li="0">ceci</li>
  <li data-numero-li="1">et cela</li>
</ul>
```

Question 9

À quoi sert l'objet `localStorage` ?

Réponse A

L'objet `localStorage` permet de stocker en mémoire du navigateur des devis réalisés avec la page de création de devis. Les devis sont conservés même si on recharge la page ou on redémarre l'ordinateur.

Réponse B

L'objet `localStorage` permet de stocker en mémoire du navigateur des données sous la forme de paires (clé, valeur). Les données sont conservées même si on recharge la page ou on redémarre l'ordinateur.

Réponse C

L'objet `localStorage` permet de stocker en mémoire du navigateur des données sous la forme de paires (clé, valeur). Les données sont conservées si on recharge la page mais on les perd si on redémarre l'ordinateur.

Réponse D

L'objet `localStorage` permet de stocker en mémoire du navigateur des données sous la forme de paires (clé, valeur). Les données sont perdues si on recharge la page ou on redémarre l'ordinateur.

Question 10

On utilise le code HTML de la page web donnée en introduction. Quel Javascript faut-il écrire dans la fonction `init()` pour tous les fils de `<article>` soient supprimés **SAUF** son titre `<h2>` ?

Réponse A

```
let article = document.querySelector('article');
while (article.childElementCount > 1) {
    article.remove(lastElementChild);
}
```

Réponse B

```
let article = document.querySelector('article');
while (article.childElementCount > 1) {
    article.lastElementChild.remove();
}
```

Réponse C

```
let article = document.querySelector('article');
while (article.childElementCount > 1) {
    remove(article.lastElementChild)
}
```

Réponse D

```
let article = document.querySelector('article');
while (article.length > 1) {
    article.lastElementChild.remove();
}
```

PARTIE 2 (10 points)

Exercice 1

On considère la page HTML de code source :

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Examen TW2 - Exercice 1</title>
  <style>
    .fond { background-color: yellow; }
  </style>
</head>
<body>

<h1>Examen TW2 - Exercice 1</h1>

<div>
  <button id="question1">Question 1</button>
  <button id="question2">Question 2</button>
  <form action="https://google.com">
    <label>Votre nom :
      <input type="text" name="nom" id="nom" value="">
    </label>
    <input type="submit" name="go" value="Afficher votre nom">
    <div id="erreur"></div>
  </form>
</div>

<p id="intro">Un paragraphe introductif de la page</p>

<section>
  <h2>Le titre de section</h2>
  <p>Un paragraphe pour la section avec de <em>l'emphase</em> et
du <strong>texte important</strong></p>
  <p>Et un <span>autre avec un span</span></p>
  <p>Et <em>enfin le <strong>dernier</strong></em></p>
</section>

<footer>
  <p>Un pied de page qui jaunit</p>
</footer>

<script>
  "use strict";
  // code JS à rédiger sur votre copie
</script>
</body>
</html>
```

Écrire le code Javascript pour :

1. Quand on clique sur le 1er bouton, le texte du paragraphe introductif devient "texte disparu"
 2. Quand on clique sur le 2e bouton, le footer appartient alors à la classe fond
 3. Au chargement de la page, les paragraphes de la section (et pas les autres) captent le clic et le texte du paragraphe cliqué devient rouge lorsqu'on clique dessus
 4. Au chargement de la page, le paragraphe d'identifiant intro capte le clic et se supprime de la page quand on clique dessus.
 5. Au chargement de la page, le formulaire capte le submit, empêche l'envoi des données chez Google et :
 - si le le champ de texte est vide, alors un message d'erreur est affiché dans la division d'identifiant erreur
 - si le texte saisi dans le champ de texte est non vide, alors un paragraphe est ajouté à la fin du <footer> avec le texte :
- Page réalisée par XXX
où XXX est le nom saisi dans le formulaire.

Exercice 2

On considère la page HTML de code source :

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Examen TW2 - Exercice 2</title>
  <script src="data.js"></script>
</head>
<body>
<h1>Examen TW2 - Exercice 2</h1>

<h2>Bibliothèque</h2>

<nav>
  <ul id="liste"></ul>
</nav>

<main></main>

<script>
  "use strict";
  init();
</script>
</body>
</html>
```

Le fichier `data.js` contient les données qui seront utilisées sur la page. Il s'agit d'une liste d'objets décrivant les livres. Une balise `<script>` placée dans l'en-tête de la page inclut le fichier `data.js` dont le contenu (partiel) est le suivant :

```
"use strict";

let bibliotheque = [
  {
    "id": "x32b",
    "titre": "La vie est un roman",
    "auteur": "Guillaume Musso",
    "prix": 22
  },
  {
    "id": "a98o",
    "titre": "Mourir sur Seine",
    "auteur": "Michel Bussi",
    "prix": 10
  },
  ...
];
```

Écrire le code Javascript pour que :

1. Au chargement de la page, la liste `` contenue dans l'élément `<nav>` sera remplie avec les titres de chaque livre, par exemple :

Bibliothèque

- La vie est un roman
- Mourir sur Seine
- Les misérables

2. Chaque élément `` capte le clic et affiche alors dans le `<main>` de la page les détails du livre, comme par exemple sur l'illustration ci-dessous.
Le `<main>` doit être vidé à chaque affichage pour ne visualiser qu'un seul livre à la fois.

Mourir sur Seine

Auteur : Michel Bussi

Prix : 10 euros