

# 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 :** \_\_\_\_\_

<b>Question 1</b>	
<b>Question 2</b>	
<b>Question 3</b>	
<b>Question 4</b>	
<b>Question 5</b>	
<b>Question 6</b>	
<b>Question 7</b>	
<b>Question 8</b>	
<b>Question 9</b>	
<b>Question 10</b>	



## 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</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 a = 14;
let b = 22;
if (a > b) {
    let c = a - b;
    console.log(c);
} else {
    let c = a + b;
    console.log(c);
}
console.log(c);
```

---

### Réponse A

36  
Uncaught ReferenceError: c is not defined

---

### Réponse B

8  
8

---

### Réponse C

36  
36

---

### Réponse D

8  
Uncaught ReferenceError: c is not defined

## Question 2

On dispose d'une liste de notes : `let notes = [ 12, 9, 16, 14, 9, 14];`

Quel code définit une fonction qui calcule le nombre de notes strictement inférieures à 10 et affiche le nombre de notes <10 de la liste notes ?

Plusieurs réponses affichent le bon résultat mais une seule est acceptable comme correcte.

---

### Réponse A

```
let nb = 0;
function nbNotes() {
  for (let i = 0; i < notes.length; i++) {
    if (notes[i] < 10) { nb++; }
  }
  return nb;
}
console.log(nbNotes());
```

---

### Réponse B

```
function nbNotes(liste) {
  let nb = 0;
  for (let i = 0; i < liste.length; i++) {
    if (liste[i] < 10) { nb++; }
  }
  return nb;
}
console.log(nbNotes(notes));
```

---

### Réponse C

```
function nbNotes(liste) {
  let nb = 0;
  for (let note in liste) {
    if (note < 10) { nb++; }
  }
  return nb;
}
console.log(nbNotes(notes));
```

---

### Réponse D

```
function nbNotes() {
  let nb = 0;
  for (let i = 0; i < notes.length; i++) {
    if (notes[i] < 10) { nb++; }
  }
  console.log(nb);
}
nbNotes();
```

### Question 3

En utilisant la page web donnée en introduction, déterminer le code placé dans la fonction `init()` qui permet de manipuler le titre `<h2>` de l'article.

---

#### Réponse A

```
document.querySelector("h2")
```

---

#### Réponse B

```
document.querySelector("titre-article")
```

---

#### Réponse C

```
document.getElementById("#titre-article")
```

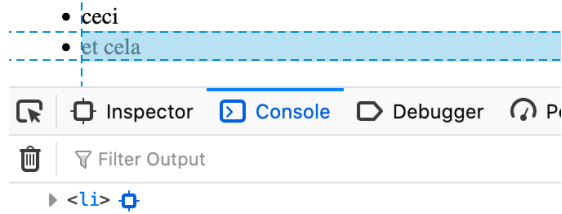
---

#### Réponse D

```
document.querySelector("#titre-article")
```

## Question 4

En utilisant la page web donnée en introduction, déterminer le code placé dans la fonction `init()` qui permet de récupérer le dernier élément `<li>` de la page et de l'afficher en console comme ci-dessous.



---

### Réponse A

```
let der = document.querySelector("li:last-child");  
console.log(der);
```

---

### Réponse B

```
let der = document.querySelectorAll("li").lastElementChild;  
console.log(der);
```

---

### Réponse C

```
let der = document.querySelectorAll("li").getLast();  
console.log(der);
```

---

### Réponse D

```
let lesLis = document.querySelectorAll("li");  
let der = lesLis[ lesLis.length - 1 ];  
console.log(der);
```

## Question 5

Quel code placé dans la fonction `init()` permet d'ajouter un élément `<p>` à la fin de l'article ?

---

### Réponse A

```
let p = document.createElement("p");  
p.textContent = "Le texte du nouveau paragraphe";  
document.querySelector("article").insert(p);
```

---

### Réponse B

```
let p = document.createElement("p");  
p.textContent = "Le texte du nouveau paragraphe";  
document.getElementById("article").appendChild(p);
```

---

### Réponse C

```
let p = document.createElement("p");  
p.textContent = "Le texte du nouveau paragraphe";  
document.querySelector("article").add(p);
```

---

### Réponse D

```
let p = document.createElement("p");  
p.textContent = "Le texte du nouveau paragraphe";  
document.querySelector("article").appendChild(p);
```



## Question 6

On utilise le code HTML de la page web donnée en introduction. Quel Javascript faut-il écrire dans la fonction `init()` pour que tous les éléments `<h2>` captent le clic et qu'un message soit affiché en console lorsque le clic est capté ?

On supposera que la fonction suivante existe :

```
function afficher() {  
    console.log("Un clic a été fait sur un h2");  
}
```

---

### Réponse A

```
let h2s = document.querySelectorAll("h2");  
for (let i = 0; i < h2s.length; i++) {  
    h2s[i].addEventListener('click', afficher());  
}
```

---

### Réponse B

```
let h2s = document.querySelectorAll("h2");  
h2s.addEventListener('click', afficher);
```

---

### Réponse C

```
let h2s = document.querySelectorAll("h2");  
h2s.addEventListener('click', afficher());
```

---

### Réponse D

```
let h2s = document.querySelectorAll("h2");  
for (let i = 0; i < h2s.length; i++) {  
    h2s[i].addEventListener('click', afficher);  
}
```

## Question 7

On utilise le code HTML de la page web donnée en introduction.

On suppose que tous les liens captent le clic et que la fonction `bloquerExt` est appelée lors du clic.

Quel Javascript faut-il écrire dans la fonction `bloquerExt` pour que si l'item `<li>` contenant le lien cliqué appartient à la classe `ext` alors il est bloqué (non activé) et un message en console s'affiche ?

---

### Réponse A

```
function bloquerExt(event) {  
  let lien = event.currentTarget;  
  if (lien.classList.contains("ext")) {  
    lien.preventDefault();  
    console.log("Ce lien est bloqué");  
  }  
}
```

---

### Réponse B

```
function bloquerExt(event) {  
  let lien = event.currentTarget;  
  if (lien.parentElement.classList.contains("ext")) {  
    lien.preventDefault();  
    console.log("Ce lien est bloqué");  
  }  
}
```

---

### Réponse C

```
function bloquerExt(event) {  
  let lien = event.currentTarget;  
  if (lien.parentElement.classList.contains("ext")) {  
    event.preventDefault();  
    console.log("Ce lien est bloqué");  
  }  
}
```

---

### Réponse D

```
function bloquerExt(event) {  
  let lien = event.currentTarget;  
  if (lien.classList.contains("ext")) {  
    event.preventDefault();  
    console.log("Ce lien est bloqué");  
  }  
}
```

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

## Question 9

À quoi sert l'objet XMLHttpRequest ?

---

### Réponse A

L'objet XMLHttpRequest permet à Javascript d'interagir avec des serveurs. On peut récupérer des données à partir d'une URL sans avoir à rafraîchir complètement la page.

---

### Réponse B

L'objet XMLHttpRequest permet à Javascript de transformer du XML en requêtes HTTP.

---

### Réponse C

L'objet XMLHttpRequest permet à Javascript d'interagir avec des serveurs en leur envoyant du XML avec le protocole HTTP.

---

### Réponse D

L'objet XMLHttpRequest permet à Javascript d'interagir avec des serveurs. On peut récupérer des données au format XML (uniquement) à partir d'une URL sans avoir à rafraîchir complètement la page.

## Question 10

On utilise le code HTML de la page web donnée en introduction. Quel Javascript faut-il écrire dans la fonction `init()` pour que tous les éléments de classe `ext` soient supprimés au chargement de la page ?

---

### Réponse A

```
document.querySelectorAll(".ext").removeAll();
```

---

### Réponse B

```
let exts = document.querySelectorAll(".ext");  
for (let i = 0; i < exts.length; i++) {  
    exts[i].remove();  
}
```

---

### Réponse C

```
let exts = document.querySelectorAll(".ext");  
for (let i = 0; i < exts.length; i++) {  
    exts[i].delete();  
}
```

---

### Réponse D

```
let exts = document.querySelectorAll(".ext");  
for (let i = 0; i < exts.length; i++) {  
    exts[i].reset();  
}
```

## PARTIE 2 (10 points)

### Exercice 1

On considère la page HTML de code source :

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Session 1 - Exercice 1</title>
</head>
<body>
  <h1>Session 1 - Exercice 1</h1>
  <div>
    <h2>Des boutons à cliquer</h2>
    <button id="question1">Question 1</button>
    <button id="question2">Question 2</button>
    <button id="question3">Question 3</button>
  </div>

  <main>
    <h2>Le titre du main</h2>
    <p>Un paragraphe pour le main 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>
  </main>
  <footer>
    <h2>Titre du footer</h2>
    <p>Un paragraphe qui disparaît</p>
    <p>Mais pas celui-ci</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 titre du <footer> devient "titre modifié"
2. Quand on clique sur le 2e bouton, le dernier paragraphe du <main> est supprimé
3. Quand on clique sur le 3e bouton, tous les paragraphes du <main> passent en couleur de texte rouge
4. Au chargement de la page, les paragraphes captent le clic et au clic le fond du paragraphe cliqué devient jaune

## Exercice 2

On veut réaliser une application de tests visuels qui consiste à afficher à l'utilisateur une suite de questions à laquelle il doit associer une image parmi 4.

Les données du questionnaire sont contenues dans une liste d'objets Javascript de type *question* sous la forme ci-dessous. La clé *reponse* correspond à l'indice de l'image correcte dans la liste des 4 images possibles.

```
let questions = [
  {
    "question": "le texte de la question",
    "images": [
      "image167.png",
      "image214.jpg",
      "image34.png",
      "image21.png"
    ],
    "reponse": "image214.jpg"
  },
  ...
];
```

L'application doit afficher une question et 4 images, et compter le nombre de bonnes réponses.

Pour cela, le HTML utilisé est le suivant :

```
<section>
  <h2>Question</h2>
  <div id="question"></div>
  <div id="images">
    
    
    
    
  </div>
</section>
<section id="resultats"></section>
```

Une fonction Javascript `init()` est appelée après le chargement de la page. Le squelette du fichier JS est le suivant :

```
"use strict";
// variable globale pour savoir où on en est dans le questionnaire
let numeroQuestion = 0;

// variable globale pour compter les bonnes réponses
let score = 0;

// fonction qui initialise le questionnaire
function init() {
  // à compléter
}
```

### Question 1

Écrire la fonction `poserQuestion(question)` qui prend en argument un objet de type `question` et affiche la question et les images dans le HTML (les éléments `<img>` étant présents dans le HTML, il suffit de changer la source de chaque image).

### Question 2

Écrire la fonction `init()` qui initialise le questionnaire en affichant la question n°0 de la liste (grâce à la fonction `poserQuestion`) et qui initialise le fait que les 4 images captent le clic. Au clic sur une image, la fonction `avancer` est appelée.

### Question 3

Écrire la fonction `avancer` qui :

- teste si l'image cliquée correspond à la bonne réponse et si oui incrémente le score  
Note : la variable globale `numeroQuestion` permet d'accéder à la question en cours dans la liste
- teste si on est arrivé au bout du questionnaire,
  - si oui, affiche alors le score dans la section résultats
  - si non, affiche la question suivante grâce à la fonction `poserQuestion`