

[Tableau de bord](#) / [Mes cours](#) / [LOG2440 - Méthod. de dévelop. et conc. d'applic. Web](#) / Contrôle pratique  
/ [LOG2440 - Automne 2021 - Contrôle Pratique](#)

Commencé le	lundi 18 octobre 2021
État	Terminé
Terminé le	lundi 18 octobre 2021
Temps mis	
Note	19,75 sur 25,00 (79%)

Description

#### LISEZ CES INSTRUCTIONS AVANT DE COMMENCER L'EXAMEN

Le contrôle pratique est composé de 6 sections. Vous êtes libres de changer de section en tout temps et de changer les réponses à vos questions. Votre contrôle pratique sera évalué seulement après avoir cliqué sur le bouton "Tout envoyer et terminer" et avoir confirmé la soumission.

Voici les règles pour l'évaluation:

- L'examen possède 14 questions notées sur un total de 25 points.
- La note partielle associée à chaque question est marquée à gauche de la question.
- Certaines questions requièrent plusieurs champs à remplir, tandis que d'autres questions sont à choix multiples.
- L'espace disponible n'est nécessairement représentatif de la taille de la réponse attendue : ne vous sentez pas obligés de remplir tout l'espace donné.

**Vous avez une seule tentative pour le contrôle pratique ! Ne soumettez pas votre tentative à moins d'être 100% sûr(e) d'avoir terminé l'examen !**

En cas de doute sur le sens d'une question, faites une supposition raisonnable, énoncez-la clairement dans votre réponse et poursuivez. Une "question" vide est disponible sur la première page d'examen si vous avez besoin de plus de place ou pour des questions spécifiques.

Les téléphones et les ordinateurs portables ne sont pas permis. Sur votre poste de travail, vous pouvez utiliser seulement Moodle.

L'ensemble des notes de cours sont disponibles sur Moodle, dans la section Ressources

Bon travail!

Question 1

Terminer

Non noté

Cette question est un espace dédié pour vos commentaires ou questions sur l'examen final. Aucune réponse ne sera donnée pendant l'examen.

Priorisez de mettre vos commentaires et/ou hypothèses directement dans la question spécifique.

Je ne vois pas l'image de la question 5, c'est donc plus difficile pour moi d'y répondre. Finalement j'ai vu au tableau, c'est bon.

## Question 2

Incorrect

Note de 0,00 sur 1,00

Quelles affirmations suivantes sont vraies par rapport à div et span ?

Veuillez choisir au moins une réponse. **Attention à votre sélection** : un mauvais choix aura un impact sur la note totale de la question.

Veuillez choisir au moins une réponse :

- ☐ a. div est un conteneur sémantique qui regroupe des éléments communs, alors que span n'offre aucune sémantique aux éléments
- ☐ b. span est une balise sectionnante, contrairement à div
- ☐ c. div et span peuvent être utilisés interchangeablement
- ☐ d. div et span ne sont pas des balises sémantiques
- ☐ e. span est une balise en-ligne, alors que div est une balise bloc
- ☒ f. div est une balise en-ligne, alors que span est une balise bloc



Les réponses correctes sont :

div et span ne sont pas des balises sémantiques,

span est une balise en-ligne, alors que div est une balise bloc

## Question 3

Correct

Note de 1,00 sur 1,00

La spécification de HTML5 a retiré la balise *center* du langage. <center> est une balise dont le contenu entier est centré horizontalement au sein de son conteneur parent.

Selon vous, pourquoi est-ce que cette balise a été retirée ?

- ☐ a. Les utilisateurs ne l'utilisaient pas de la façon définie dans la spécification de HTML4
- ☐ b. Le nom de la balise portait à confusion
- ☐ c. Le centrage de la balise offre une très haute spécificité CSS, donc on ne peut pas la redéfinir dans une feuille de style CSS
- ☒ d. La balise joue un rôle de mise en forme plutôt qu'un rôle de structure



La réponse correcte est :

La balise joue un rôle de mise en forme plutôt qu'un rôle de structure

## Question 4

Correct

Note de 1,00 sur 1,00

Voici un extrait de la fonction "generateQuestion" permet de générer le contenu HTML d'une question sur une plateforme de quiz en ligne.

```
1 function generateQuestion() {  
2   const question = document.createElement("div");  
3   question.dataset.questionId = "id-1";  
4   question.dataset.category = "category-HTML";  
5   question.dataset.multipleChoice = "false";  
6   // reste de la configuration  
7   document.body.appendChild(question);  
8 }
```

Suite à l'exécution de la fonction, quel sera le contenu de l'élément <div> généré par la fonction ?

- ☐ a. <div data-questionId="id-1" data-category="category-HTML" data-multipleChoice="false"></div>
- ☐ b. Aucun contenu. La majuscule dans "category-HTML" rend le code erroné : pas de majuscules après un tiret dans un dataset.
- ☒ c. <div data-question-id="id-1" data-category="category-HTML" data-multiple-choice="false"></div> ✓
- ☐ d. <div data-question-Id="id-1" data-category="category-HTML" data-multiple-Choice="false"></div>
- ☐ e. <div dataset-question-id="id-1" dataset-category="category-HTML" dataset-multiple-choice="false"></div>
- ☐ f. Aucun contenu. Le tiret dans "id-1" et "category-HTML" rend le code erroné : les tirets sont réservés pour les noms des attributs du dataset.

La réponse correcte est :

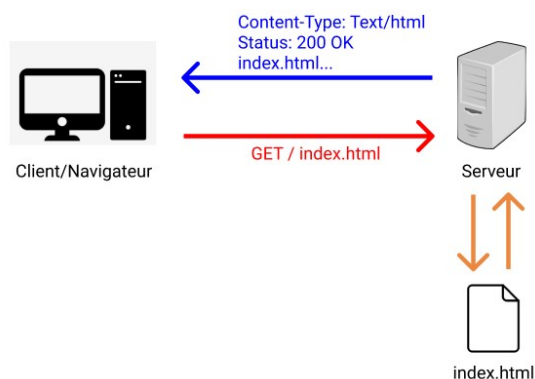
<div data-question-id="id-1" data-category="category-HTML" data-multiple-choice="false"></div>

## Question 5

Correct

Note de 1,00 sur 1,00

Décrivez le scénario correct en se basant sur l'image ci-dessous.



- ☐ a. Le serveur demande un fichier avec le contenu text/html nommée index.html au client qui renvoie cette page en question au serveur.
- ☒ b. Le client fait une requête HTTP en précisant une méthode HTTP GET /index.html au serveur. Ce dernier reçoit et traite la requête et répond au client avec les informations demandées (index.html) et un code HTTP 200 OK. ✓
- ☐ c. Le client envoie la page index.html au serveur à travers la méthode HTTP /GET. Le serveur s'en occupe pour sauvegarder cette page et répond avec un code HTTP 200 OK.
- ☐ d. Le serveur envoie une requête HTTP /GET et le navigateur répond avec un code HTTP ainsi que la page index.html.

La réponse correcte est :

Le client fait une requête HTTP en précisant une méthode HTTP GET /index.html au serveur. Ce dernier reçoit et traite la requête et répond au client avec les informations demandées (index.html) et un code HTTP 200 OK.

## Question 6

Correct

Note de 1,00 sur 1,00

Quelques collègues et vous, avez décidé de partir une entreprise en consultation de développement logiciel. Vous avez mis en place un site web pour votre compagnie PolyConsulte . Cependant, depuis ce matin, vous recevez des courriels comme quoi votre site web n'est plus accessible à travers polyconsulte.com , mais l'accès direct à l'adresse IP de votre serveur (132.207.12.43) reste fonctionnel.

Où peut être le problème ?

- ☐ a. Les utilisateurs ne rentrent pas le bon nom de domaine du site
- ☐ b. Le serveur HTTP est en panne
- ☒ c. La configuration du service DNS utilisé pour accéder au site ✓
- ☐ d. Votre webmestre est un ancien stagiaire de Facebook

La réponse correcte est :

La configuration du service DNS utilisé pour accéder au site

## Question 7

Correct

Note de 1,00 sur 1,00

À quoi correspondent les valeurs de chaque côté de l'élément *box* selon le code suivant?

*Note: Il est recommandé d'utiliser la même unité pour la même propriété.*

```
1 .box {  
2   padding: 10rem 20px 8vw;  
3 }
```

☐ a.

```
1 padding-top: 10rem;  
2 padding-right: 20px;  
3 padding-bottom: 20px;  
4 padding-left: 8vw;
```

☐ b.

```
1 padding-top: 8vw;  
2 padding-right: 20px;  
3 padding-bottom: 20px;  
4 padding-left: 10rem;
```

☒ c.

```
1 padding-top: 10rem;  
2 padding-right: 20px;  
3 padding-bottom: 8vw;  
4 padding-left: 20px;
```



☐ d.

```
1 padding-top: 8vw;  
2 padding-right: 20px;  
3 padding-bottom: 10rem;  
4 padding-left: 20px;
```

La réponse correcte est :

```
1 padding-top: 10rem;  
2 padding-right: 20px;  
3 padding-bottom: 8vw;  
4 padding-left: 20px;
```

## Question 8

Terminer

Note de 0,00 sur 2,00

Est-ce que les deux sélecteurs CSS suivants pourraient s'appliquer au même élément :

**div \* p**

**div > p**

Si oui, **expliquez pourquoi** en donnant un **exemple** de code HTML. Si non, **expliquez pourquoi**.

Non, c'est impossible.

**div > p**, permet de sélectionner tous les enfants direct des div qui sont p (donc on sélectionne ici que des enfants directs du div) (donc pas les petits enfants)

Alors que **div \* p** sélectionne tous les p dans une autre balise quelconque dans des div. Ceci sélectionne forcément des petits enfants du div (donc pas d'enfants)

## Question 9

Correct

Note de 1,00 sur 1,00

Les bonnes pratiques de créations de pages avec HTML et CSS proposent de séparer les deux, donc aucune utilisation de l'attribut style dans le HTML.

Parmi les affirmations suivantes, laquelle ou lesquelles justifient cette pratique ?

- ☒ a. Permet d'avoir plusieurs mise en forme pour plusieurs tailles d'écran
- ☒ b. Meilleure gestion de la spécificité des sélecteurs utilisés
- ☒ c. Évite la duplication de code CSS.
- ☒ d. Permet de changer la mise en forme de la page sans affecter le contenu



Les réponses correctes sont :

Évite la duplication de code CSS.,

Permet de changer la mise en forme de la page sans affecter le contenu,

Meilleure gestion de la spécificité des sélecteurs utilisés,

Permet d'avoir plusieurs mise en forme pour plusieurs tailles d'écran

## Description

Voici un annexe de fonctions/méthodes Javascript et du DOM

## Fonctions et méthodes Javascript

Fonction/Méthode	Description
.length	Retourne la taille du tableau
.indexOf(valeur)	Retourne l'indice du tableau qui contient la valeur passée en paramètre. Sinon -1.
.push(valeur)	Ajoute un élément au tableau
.slice(begin, end)	Retourne un nouveau tableau à partir de la case à l'indice <i>begin</i> (obligatoire) jusqu'à l'indice <i>end</i> (optionnel).
.map(fonction)	Applique une fonction de transformation sur chaque élément du tableau et retourne un nouveau tableau
.filter(predicat)	Filtre les éléments du tableau en fonction du prédicat et retourne le résultat dans un nouveau tableau
.find(predicat)	Recherche le premier élément qui satisfait le prédicat. Sinon, retourne <i>undefined</i> .
.forEach(fonctionParc)	Applique la fonction <i>fonctionParc</i> sur chaque élément d'un tableau. Ne retourne rien
.sort(comparateur)	Retourne un nouveau tableau triée selon la fonction <i>comparateur</i>
Object.keys(obj)	Retourne toutes les clés de l'objet Javascript <i>obj</i>

## Propriétés et méthodes de l'objet *document*

Fonction/Méthode	Description
getElementById()	Retourne l'élément possédant le ID passé en paramètre.
getElementsByName()	Retourne une collection contenant tous les éléments dont la balise correspond à celle passée en paramètre.
getElementsByClassName()	Retourne une collection contenant tous les éléments dont la classe correspond à celle passée en paramètre.
querySelector()	Retourne le premier élément correspondant au sélecteur CSS passé en paramètre.
querySelectorAll()	Retourne une collection contenant tous les éléments correspondant au sélecteur CSS passé en paramètre.
createElement()	Crée un élément correspondant à la balise passée en paramètre.
createTextNode()	Crée un nouveau noeud texte.

## Propriétés et méthodes associées à un noeud *N* du DOM

Fonction/Méthode	Description
innerHTML	Contient tout le contenu HTML inclus dans un noeud.
childNodes	Collection contenant les noeuds fils, incluant les noeuds texte.
children	Collection contenant les noeuds fils, excluant les noeuds texte.
firstChild	Retourne le premier noeud fils
firstElementChild	Retourne le premier fils, en ne considérant pas les noeuds texte.
lastChild	Retourne le dernier noeud fils
lastElementChild	Retourne le dernier fils, en ne considérant pas les noeuds texte.
nextSibling	Retourne le noeud qui a le même parent que N et qui est situé immédiatement après.
nextElementSibling	Retourne le noeud qui a le même parent que N et qui est situé immédiatement après, en ne considérant pas les noeuds texte.
previousSibling	Retourne le noeud qui a le même parent que N et qui est situé immédiatement avant.
previousElementSibling	Retourne le noeud qui a le même parent que N et qui est situé immédiatement avant, en ne considérant pas les noeuds texte.
parentNode	Retourne le noeud parent.
textContent	Retourne tout le contenu textuel d'un noeud (incluant ses descendants).



Fonction/Méthode	Description
nodeType	Retourne le type d'un noeud. En particulier : 1 = noeud élément, 2 = noeud attribut, 3 = noeud texte.
style	Permet d'obtenir et de changer le style d'un élément. Exemple : monElement.style.color = blue.
className	Permet d'obtenir et de changer la classe d'un élément.
addEventListener(ev, gest, cap)	Associe un gestionnaire d'événement à un événement. Si le troisième argument est true le traitement se fait aussi au cours de la capture (pas seulement lors de la propagation vers le haut (bubbling)).
appendChild()	Ajoute le noeud passé en paramètre après tous les noeuds fils de N .
insertBefore(nouv, ref)	Ajoute le noeud nouv juste avant le noeud ref , qui est un noeud fils de N .
remove()	Retire l'élément de l'arbre DOM

## Question 10

Correct

Note de 4,00 sur 4,00

Implémentez la fonction *filter* qui permet de filtrer un tableau Javascript. La fonction demandée retourne tous les éléments qui satisfont un prédicat passé en paramètre, sinon renvoie un tableau vide.

## CONTRAINTES:

Vous ne pouvez pas utiliser la méthode *filter* existante, car on vous demande de l'implémenter.

Respectez le principe d'immuabilité.

Des points seront donnés en fonction de l'élégance du code.

Vous avez autant d'essais que nécessaires pour faire fonctionner le code.

## Par exemple:

Test	Résultat
<code>console.log(filter([1,4,3], x =&gt; x &gt; 2))</code>	<code>[ 4, 3 ]</code>
<code>console.log(filter([1,4,3], x =&gt; x &gt; 5))</code>	<code>[]</code>

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
/**
 * Applique un prédicat sur chaque élément d'un tableau et renvoie un tableau
 * avec tous les éléments qui satisfont le prédicat. Sinon, renvoie undefined.
 *
 * @example
 * filter([1,4,3], x => x > 2) // renvoie [4,3]
 * @example
 * filter([1,4,3], x => x > 5) // renvoie undefined
 *
 * @param {Array} arr - Tableau sur lequel appliquer le filtre
 * @param predicate - Prédicat de filtrage
 * @returns {T} un tableau les éléments trouvés ou undefined si rien n'est trouvé
 */
function filter(arr, predicate) {
  let t = [];
  let nbAjout = 0;

  if(arr === undefined){
```

	Test	Résultat attendu	Résultat obtenu	
✓	<code>console.log(filter([1,4,3], x =&gt; x &gt; 2))</code>	<code>[ 4, 3 ]</code>	<code>[ 4, 3 ]</code>	✓
✓	<code>console.log(filter([[1, 2], [3, 4], [5, 6, 7]], x =&gt; x.length === 3))</code>	<code>[ [ 5, 6, 7 ] ]</code>	<code>[ [ 5, 6, 7 ] ]</code>	✓
✓	<code>console.log(filter(undefined, v =&gt; v.length === 3))</code>	<code>[]</code>	<code>[]</code>	✓
✓	<code>console.log(filter([]))</code>	<code>[]</code>	<code>[]</code>	✓
✓	<code>console.log(filter([1,4,3], x =&gt; x &gt; 5))</code>	<code>[]</code>	<code>[]</code>	✓

	Test	Résultat attendu	Résultat obtenu	
✓	<pre>const fruits = ['apple', 'banana', 'grapes', 'mango', 'orange'] const filterItems = (arr, query) =&gt; {   return arr.filter(el =&gt; el.toLowerCase().indexOf(query.toLowerCase())     !== -1) }  console.log(filterItems(fruits, 'ap'))</pre>	[ 'apple', 'grapes' ]	[ 'apple', 'grapes' ]	✓

Tous les tests ont été réussis ! ✓

### Solution de l'auteur de la question (Nodejs):

```

1  /**
2   * Applique un prédicat sur chaque élément d'un tableau et renvoie
3   * avec tous les éléments qui satisfont le prédicat. Sinon, renvoie
4   *
5   * @example
6   * filter([1,4,3], x => x > 2) // renvoie [4,3]
7   * @example
8   * filter([1,4,3], x => x > 5) // renvoie undefined
9   *
10  * @param {Array} arr - Tableau sur lequel appliquer le filtre
11  * @param predicate - Predicat de filtrage
12  * @returns {T} un tableau les éléments trouvés ou undefined si rien
13  */
14  function filter(arr, predicate) {
15    const res = [];
16    if (!arr || arr.length === 0) {
17      return res;
18    }
19    else{
20      arr.forEach(x=>{
21        if(predicate(x)){
22          res.push(x);
23        }
24      })
25    }
26  }

```

Correct

Note pour cet envoi : 4,00/4,00.

## Question 11

Terminer

Note de 2,00 sur 2,00

Voici l'implémentation d'une fonction nommée **log2440**.

```
1 const log2440 = arr => {  
2   return arr.reduce((acc, x) => {  
3     if (acc.includes(x)) {  
4       return acc;  
5     }  
6     return acc.concat([x]);  
7   }, [])  
8 }
```

a) Expliquez ce que la fonction fait. Proposez un meilleur nom pour la fonction. (1.5 points)

b) Quelle sera la valeur de retour si on appelle **log2440** avec [1, 3, 2, 2, 1] ? (0.5 points)

a) La fonction supprime les doublons du tableau qu'on lui passe en paramètre (et retourne le tableau filtré ensuite). Un meilleur nom pour cette fonction serait "retirerDoublons".

b) Cela retournera [1, 3, 2]

## Question 12

Correct

Note de 1,00 sur 1,00

Voici un extrait de code JavaScript. La sortie de la console à la ligne 10 est {} (Objet vide). Pourquoi ?

```
1 function Car(){
2   return {
3     name: "Honda",
4     make : "Civic",
5     year : 2004
6   }
7 }
8 const car = Car();
9 const honda = Object.create(car);
10 console.log(honda); // {}
```

- ☐ a. Il manque le mot clé "new" lors de l'appel de Car() à la ligne 8
- ☐ b. Les attributs "name", "make" et "year" sont sur le Prototype de l'objet "car"
- ☐ c. C'est faux, la ligne 10 affichera { name: 'Honda', make: 'Civic', year: 2004} dans la console
- ☐ d. Object.create() ne fonctionne pas avec un objet qui n'est pas créé par une fonction constructeur
- ☐ e. Object.create() retourne un nouveau objet vide
- ☒ f. Les attributs "name", "make" et "year" sont sur le Prototype de l'objet "honda" ✓

Les réponses correctes sont :

Les attributs "name", "make" et "year" sont sur le Prototype de l'objet "honda",

Object.create() retourne un nouveau objet vide

## Question 13

Terminer

Note de 2,25 sur 2,50

Les objets suivants retournent des références vers les mêmes éléments du DOM.

a) **Quelles sont les différences** entre les 2 objets (byCN et byQS) ?

```
const byCN = document.getElementsByClassName("recette");
```

```
const byQS = document.querySelectorAll(".recette");
```

b) Les 2 objets sont déclarés comme "**const**". Est-ce qu'il aura une modification dans un ou les deux objets si j'ajoute un élément avec la classe "recette" à ma page web. Si oui, **lequel(s)** et **pourquoi** ? Si non, **pourquoi** ? **Justifiez** votre réponse.

a) byQS est statique (pas vivant) contrairement à byCN. De plus on peut accéder aux éléments de byCN comme si c'était un tableau (contrairement à byQS)

b) Oui, un des deux sera quand même modifier, il s'agit de byCN, étant donné qu'il est vivant (pas statique). byQS ne sera pas modifié étant donné qu'il est statique.

## Question 14

Terminer

Note de 1,50 sur 2,50

Donner 2 manières différentes de retirer un gestionnaire d'événement d'un élément HTML.

Donnez un **exemple** et un **avantage** d'utilisation pour chacune de vos réponses.

Méthode 1) `removeEventListener(type,listener[, options])` (donc tenter de retirer le listener avec la méthode `removeEventListener`)

Avantage, s'assure qu'il ne sera pas possible (par ce moyen) de retirer le listener dans les cas où il est défini comme une expression de fonction.

Exemple:

```
addEventListener(onLoad(), console.log(allo));  
removeEventListener(onLoad(), console.log(allo));
```

Méthode 2) utiliser la méthode `cloneNode()`

Avantage, permettre de supprimer l'événement même s'il a été déclaré avec des expressions de fonctions.

Exemple:

```
const myNode = document.getElementById(myNode);  
addEventListener(onLoad(), console.log(allo));  
myNode = cloneNode(myNode);
```

## Question 15

Terminer

Note de 3,00 sur 4,00

a) Voici un extrait des tests unitaires pour la fonction `calculateDistance` de la classe `DistanceService`. En fonction des tests présentés, implémentez la fonction.

```
1 describe("Calculate distance tests", () => {
2   let start, end, distance;
3   let service;
4   beforeEach(() => {
5     service = new DistanceService();
6     start = { x: 0, y: 0 };
7     end = { x: 3, y: 4 };
8     distance = 5;
9   });
10
11   it("should calculate distance between two points", () => {
12     const calculatedDistance = service.calculateDistance(start, end);
13
14     expect(calculatedDistance).toBe(distance);
15   });
16
17   it("should always have a positive distance", () => {
18     end = { x: -3, y: -4 };
19     const calculatedDistance = service.calculateDistance(start, end);
20
21     expect(calculatedDistance).toBeGreaterThanOrEqual(0);
22     expect(calculatedDistance).toBe(distance);
23   });
24
25   it("should send a warning if distance is 0 ", () => {
26     start = { x: 3, y: 4 };
27     const spy = jest.spyOn(service, "sendWarning").mockImplementation(() => {});
28
29     service.calculateDistance(start, end);
30     expect(spy).toHaveBeenCalled();
31   });
32 });
```

b) Quel est le fonctionnement de la ligne suivante ? Est-ce que l'utilisation de "mockImplementation" est adéquate ici ? Justifiez votre réponse

```
1 const spy = jest.spyOn(service, "sendWarning").mockImplementation(() => {});
```

a)

```
class DistanceService {
  calculateDistance(let start, let end) {
    if(start.x===end.x && start.y===start.y){
      new Event("sendWarning");
      console.log("WARNING : distance = 0");
    }
    const distance = Math.sqrt( (end.x-start.x)*(end.x-start.x) + (end.y-start.y)*(end.y-start.y) )
    return distance;
  }

  // Reste de la classe
};
```

b) Non, elle n'est pas adéquate. Je crois qu'il est nécessaire de recréer l'évènement dans la classe de test pour pouvoir utiliser la fonction "toHaveBeenCalled"



Aller à...

[LOG2440 - Automne 2021 - Examen Final](#) ►