

ÉPREUVE THÉORIQUE HTML-CSS-JAVASCRIPT

Session	Février 2025
Spécialités	DA

Durée	5h
Coefficient	5
Note éliminatoire	12/20

**Partie 1 : QRO.....01heure ; Partie 2 : QRO.....02heures ;
Partie 3 : QCM et justifier02heures**

Partie 1 : Questions1heure

Exercice 1: Notions de base en HTML et CSS....5 points

Vous êtes un développeur web travaillant sur un nouveau projet de site internet. Vous devez comprendre et utiliser les balises HTML pour structurer et organiser le contenu de vos pages web. Une balise HTML est une instruction utilisée pour marquer et définir différents éléments dans une page web. Par exemple, la balise <h1> est couramment utilisée pour définir un titre principal dans une page. Elle permet de communiquer aux navigateurs web la structure et la signification des différents éléments présents dans le code HTML

1. a) Définition : Balise HTML, attribut
b) Donnez un exemple de balise HTML couramment utilisée.
2. a) Définition : Sélecteur CSS
b) Donnez un exemple de sélecteur CSS couramment utilisé.
3. a) Définition : Propriété CSS
b) Comment est-elle utilisée pour appliquer des styles à un élément HTML ?
4. a) Définition : Classe CSS
b) Comment est-elle utilisée pour regrouper des éléments HTML et leur appliquer des styles communs ?
c) Donnez un exemple d'utilisation d'une classe CSS dans une page web.

Exercice 2 : Tables, listes et formulaire....5 points

Question 1 : Expliquez comment créer une table HTML avec trois colonnes et deux lignes, où chaque cellule contient un contenu spécifique. Incluez les balises nécessaires et décrivez la structure requise.

Question 2 : Décrivez comment créer une liste non ordonnée avec cinq éléments, chacun contenant un paragraphe de texte. Utilisez les balises appropriées et indiquez comment structurer correctement la liste dans le code HTML.

Question 3 : Vous souhaitez créer un formulaire d'inscription pour un événement. Expliquez comment ajouter des champs de saisie de texte pour le nom, l'adresse e-mail et le numéro de téléphone. Incluez les balises HTML nécessaires et indiquez comment spécifier les attributs appropriés pour chaque champ.

Question 4 : Vous avez besoin de créer une liste ordonnée numérotée de dix éléments dans un document HTML. Expliquez comment utiliser les balises HTML adéquates pour obtenir cette liste, en commençant par le numéro 1 et en allant jusqu'au numéro 10.

Question 5 : Vous souhaitez afficher une image dans une cellule spécifique d'un tableau HTML. Décrivez comment accomplir cela en utilisant les balises HTML appropriées, en spécifiant l'image source et en ajustant éventuellement les dimensions de l'image.

Exercice 3 : Éléments interactifs.....5 points

Situation : Vous développez une application web pour un restaurant qui permet aux clients de commander des plats en ligne. Vous devez créer des éléments interactifs pour améliorer l'expérience utilisateur.

Question 1 : Vous souhaitez ajouter un bouton "Ajouter au panier" à chaque plat affiché sur la page de menu. Comment pouvez-vous créer ce bouton en utilisant les éléments HTML appropriés ?

Question 2 : Pour faciliter la navigation des clients dans le menu, vous décidez d'ajouter un menu déroulant permettant de filtrer les plats par catégorie (entrées, plats principaux, desserts, etc.). Comment pouvez-vous créer ce menu déroulant en utilisant les éléments HTML adéquats ?

Question 3 : Vous voulez donner aux clients la possibilité d'ajouter des commentaires ou des instructions spéciales lorsqu'ils passent leur commande. Comment pouvez-vous créer un champ de texte où les clients peuvent entrer ces informations en utilisant les éléments HTML appropriés ?

Question 4 : Vous souhaitez afficher une image agrandie d'un plat lorsque le client passe la souris dessus. Comment pouvez-vous réaliser cela en utilisant les éléments HTML et les événements appropriés ?

Question 5 : Pour offrir une expérience interactive plus fluide, vous souhaitez afficher un message de confirmation chaque fois qu'un plat est ajouté au panier. Comment pouvez-vous mettre en place cette fonctionnalité en utilisant les éléments HTML et les scripts JavaScript appropriés ?

Exercice 4 : Introduction au CSS3.....2 points

Situation de vie : Vous souhaitez styliser votre page web en utilisant les fonctionnalités avancées de CSS3.

1. Comment spécifier les dimensions d'un élément en CSS3 ?
2. Comment appliquer des couleurs à un élément en CSS3 ?

Thème : Positionnement avec CSS..... 5 points

1. Quels sont les différents types de positionnement disponibles en CSS ?
2. Comment positionner un élément de manière absolue par rapport à son conteneur parent ?
3. Comment centrer horizontalement et verticalement un élément en utilisant le positionnement absolu ?
4. Qu'est-ce que le positionnement fixe et comment peut-il être utilisé dans la création d'un élément de navigation fixe en haut de la page ?
5. Quelle est la différence entre le positionnement relatif et absolu en CSS ?

Thème : Media Queries et Responsive Design.....5 points

1. Qu'est-ce qu'une media query en CSS et comment est-elle utilisée pour créer un design responsive ?

2. Comment définir différents styles CSS pour différentes résolutions d'écran à l'aide des media queries ?
3. Quelles sont les bonnes pratiques à suivre pour rendre un site web responsive ?
4. Comment rendre une image réactive en utilisant des media queries ?
5. Comment cacher ou afficher certains éléments en fonction de la taille de l'écran à l'aide des media queries ?

Thème : Flexbox.....5 points

1. Qu'est-ce que Flexbox et comment fonctionne-t-il en CSS ?
2. Comment créer un conteneur flex et définir les propriétés de flexbox pour aligner les éléments ?
3. Quelles sont les propriétés CSS utilisées pour aligner les éléments en ligne et en colonne dans un conteneur flex ?
4. Comment modifier l'ordre d'affichage des éléments dans un conteneur flex à l'aide de la propriété order ?

Partie 2 : Questions2heures

Exercice complet : JavaScript - Notions de base.....5 points

Situation de problème : Vous avez été engagé en tant que développeur JavaScript pour créer une application de gestion de tâches. Votre mission consiste à utiliser les notions de base de JavaScript pour résoudre les problèmes suivants :

1. Introduction au JavaScript :
définir script et expliquez en quelques phrases ce qu'est le JavaScript et son rôle dans le développement web.
2. Configuration de l'environnement de développement :
 - a) Citez 5 types de variables en JS.
 - b) Citez les méthodes pour insérer le code JavaScript dans une page HTML et expliquez quelle est la meilleure méthode.

3. Structures de contrôle :
 - a) Écrivez une structure de contrôle conditionnelle qui vérifie si un nombre est pair ou impair.
 - b) Utilisez la console pour afficher le résultat.
4. Fonctions :
 - a) Créez une fonction appelée "calculerSurfaceRectangle" qui prend deux paramètres (longueur et largeur) et renvoie la surface du rectangle.

Exercice 1 : Manipulation du DOM et événements.....5 points

Vous travaillez sur une application web pour un site de réservation de billets de cinéma. Vous devez utiliser les notions suivantes pour réaliser les fonctionnalités demandées :

1. Qu'est-ce que le DOM ?
 - a) Expliquez en quelques phrases ce qu'est le DOM (Document Object Model) et son rôle dans le développement web.
2. Les sélecteurs :
 - a) Utilisez un sélecteur pour sélectionner un élément avec l'id "titre" dans le DOM.
 - b) Modifiez le contenu de cet élément pour afficher le titre du film.
3. AddEventListener :
 - a) Ajoutez un événement de clic à un bouton avec l'id "btn-reserver".
 - b) Lorsque le bouton est cliqué, appelez une fonction qui affiche un message de confirmation de réservation.
4. Types d'événements :
 - a) Citez trois types d'événements couramment utilisés dans le développement web.
 - b) Donnez un exemple d'utilisation pour chaque type d'événement.

Exercice 2 : Manipulation du DOM5pts

Contexte : Vous travaillez sur une application de calculatrice en ligne. Vous devez utiliser les notions suivantes pour créer les fonctionnalités de la calculatrice :

1. Manipulation du DOM :
 - a) Sélectionnez l'élément avec l'id "resultat" dans le DOM.
 - b) Modifiez le contenu de cet élément pour afficher le résultat des calculs.

2. Utilisation de forEach :

- a) Déclarez un tableau contenant une liste de nombres.
- b) Utilisez la méthode forEach pour itérer sur chaque élément du tableau et afficher leur valeur dans la console.

Exercice sur les événements de souris et de clavier dans une situation de vie : 5pts

Situation : Vous développez un jeu en ligne où les joueurs doivent attraper des objets qui tombent du haut de l'écran en utilisant leur souris et leur clavier. Vous devez implémenter des événements de souris et de clavier pour rendre le jeu interactif.

Question 1 : Dans le jeu, comment pouvez-vous détecter le clic de souris sur un objet lorsqu'un joueur essaie de l'attraper ? Expliquez comment vous pouvez utiliser un événement de souris pour effectuer cette détection.

Question 2 : Pour permettre aux joueurs de déplacer leur personnage à l'aide des touches fléchées du clavier, comment pouvez-vous écouter les événements de pression de touches correspondantes ? Décrivez comment vous pouvez associer ces événements à des actions de déplacement dans le jeu.

Question 3 : Lorsqu'un joueur réussit à attraper un objet, vous souhaitez afficher un message de félicitations. Comment pouvez-vous détecter cet événement de réussite et afficher le message en utilisant les événements de souris appropriés ?

Question 4 : Pour ajouter une fonctionnalité supplémentaire, vous voulez permettre aux joueurs de mettre le jeu en pause en appuyant sur la touche "Espace" du clavier. Comment pouvez-vous détecter cet événement de pression de touche et mettre en pause le jeu en conséquence ? Expliquez comment vous pouvez gérer cet événement.

Partie 3 : QCM, puis justifier.....2heures

1. Quelle est la sortie du code suivant ?

```
console.log(a);
```

```
var a = 5;
```

- a) undefined
- b) 5
- c) ReferenceError
- d) null

2. Lequel des éléments suivants est vrai

- a) Les déclarations de fonction sont hoistées avant les déclarations de variables.
- b) Les expressions de fonction ne sont pas hoistées.
- c) Les deux a et b sont vrais.
- d) Les fonctions fléchées ne sont pas hoistées.

3. Que se passe-t-il si vous essayez d'accéder à une variable déclarée avec let avant sa déclaration ?

```
console.log(b);
```

```
let b = 10;
```

- a) undefined
- b) 10
- c) ReferenceError
- d) null

4. Quel est le résultat du code suivant ?

```
foo();
```

```
function foo() {  
  console.log("Hello, World!");  
}
```

- a) Hello, World!
- b) undefined
- c) ReferenceError
- d) TypeError

5. Soit le script suivant

```
const a = 'Myra';  
const b = 'Myra';  
  
if (a === b) {  
  console.log('TRUE');  
} else {  
  console.log('FALSE');  
}
```

- a) TRUE**
- b) FALSE**

6. Soit le script suivant

```
const a = { username: 'Myra' };  
const b = { username: 'Myra' };  
if (a.username === b.username) {  
  console.log('TRUE');
```

```
} else {  
  console.log('FALSE');  
}
```

a) TRUE

b) FALSE

7. Soit le script suivant

```
const a = { username: 'Myra' };  
const b = { username: 'Myra' };  
if (a === b) {  
  console.log('TRUE');  
} else {  
  console.log('FALSE');  
}
```

a) TRUE

b) FALSE

8. Soit le script suivant

```
const a = ['Myra'];  
const b = ['Myra'];
```

```
if (a === b) {  
  console.log('TRUE');  
} else {  
  console.log('FALSE');  
}
```

a) TRUE

b) FALSE

9. Soit le script suivant

```
const a = "10";  
const b = 10;  
if (a == b) {  
  console.log('TRUE');  
} else {  
  console.log('FALSE');  
}
```

a) TRUE

b) FALSE

10. Soit le script suivant

```
const valeur1 = { "current": 0 };  
const valeur2 = valeur1;  
const valeur3 = valeur2;  
valeur3.current = 1;  
console.log({ valeur1, valeur2, valeur3 });
```

- a) Valeur 1=0 ; valeur 2=0 ; valeur 3=1
- b) Valeur1=1 ; valeur 2=0 ; valeur3=1
- c) Valeur 1=102 ; valeur 2=1 ; valeur3=1

11. Soit le script suivant

```
let valeur1 = 0;  
let valeur2 = valeur1;  
let valeur3 = valeur2;  
valeur3 = 102;  
console.log({ valeur1, valeur2, valeur3 });
```

- a) Valeur 1=0 ; valeur 2=102 ; valeur 3=0 ;
- b) Valeur1=0 ; valeur 2=0 ; valeur3=102 ;
- c) Valeur 1=102 ; valeur 2=0 ; valeur3=0 ;

12. À quoi sert le mot-clé **return** ?

- a) Arrête la fonction

- b) Retourne une fonction
- c) Retourne une variable
- d) Retourne 'rien' (undefined)
- e) Throw une erreur

13. Ce code implémente-t-il le pattern de **early return** ?

JS

```
const verifyUser = (user) => {  
  if (user.name === "John") {  
    if (user.age > 18) {  
      if (user.email) {  
        return true;  
      }  
    }  
  }  
  return false;  
}
```

- a) Non
- b) Oui

14. Quelle valeur sera affichée dans la console ?

JS

```
const demo = (v) => {
```

```
if (v > 4) {  
  return "Hello";  
  if (v > 5){  
    return "World";  
  }  
}  
  
console.log(demo(10));
```

- a)** Hello
- b)** World
- c)** Hello World
- d)** Rien

15.C'est possible ?

JS

```
const names = ["Rostand", "Eva", "Myra"];  
names = ["Jean"]
```

- a)** Oui
- b)** Non

16.Soit le script suivant

```
let valeur1 = 0;
```

```
let valeur2 = valeur1;
```

```
let valeur3 = valeur2;
```

```
valeur3 = 102;
```

```
console.log({ valeur1, valeur2, valeur3 });
```

a) Valeur 1=0 ; valeur 2=102 ; valeur 3=0 ;

b) Valeur1=0 ; valeur 2=0 ; valeur3=102 ;

c) Valeur 1=102 ; valeur 2=0 ; valeur3=0 ;

17.Ce tableau est-il valide ?

JS

```
const names = ["Myra", 1, false];
```

a) Oui

b) Non

18.**brands** est-il égal à **rest** ?

a) Oui

b) Non

19.Comment nommer un tableau qui représente la liste de tous mes produits ?

a) product

- b)** productList
- c)** productsList
- d)** products
- e)** productsArray
- f)** productsTableau

20. Quelle est la valeur spéciale en JavaScript qui signifie 'Pas un Nombre' ?

- a)** null
- b)** undefined
- c)** NaN
- d)** False

Explication

22. Que renvoie la méthode Number.isNaN() si la valeur convertie en Number n'est pas un nombre ?

- a)** true
- b)** false
- c)** NaN
- d)** undefined

23. La condition est true ou false :

```
if (0) {  
    // si la condition est vraie  
} else {  
    // si la condition est fausse  
}
```

a) true

b) false

24. Vrai ou faux...

JS

```
const variable = 4;  
if (variable > 3 && variable < 5) {  
    // ...  
}
```

a) vrai

b) faux

25. Vrai ou faux...

JS

```
const variable = 4;  
if (variable && variable > 3) {  
  // ...  
}
```

a) vrai

b) faux

26. Vrai ou faux...

JS

```
const variable = 4;  
if (variable && variable < 3) {  
  // ...  
}
```

a) vrai

b) faux

27. Vrai ou faux...

JS

```
const variable = 1;  
if (variable > 3 || variable < 5) {  
  // ...  
}
```

a) vrai

b) faux

28. Vrai ou faux...

JS

```
const variable = 4;  
if (variable === 3 || variable < 3 || variable > 1) {  
  // ...  
}
```

a) vrai

b) faux

29. Vrai ou faux...

JS

```
const variable = 4;  
if (!variable && variable > 3) {  
  // ...  
}
```

a) vrai

b) faux

30. Vrai ou faux...

JS

```
const variable = 4;  
if (!variable || variable > 3) {  
  // ...  
}
```

a) vrai

b) faux

31. Vrai ou faux...

JS

```
const variable = 104;  
if (variable || (variable > 1 && variable !== 104)) {  
  // ...  
}
```

- a)** vrai
- b)** faux

32. Quel est le résultat ?

TS

```
console.log(4 == "4" || 4 === "4");
```

- a)** true
- b)** false

33. Quel signe privilégier pour faire une condition ?

- a)** ==
- b)** ===

34. Comment traduire cette condition :

JS

```
if (name.length > 5 || name.length > 2 && name[0] === "B")
```

- a) Si name possède plus de 5 caractères et qu'il possède plus de 2 caractères et qu'il commence par B
- b) Si name possède plus de 5 caractères ou qu'il possède plus de 2 caractères et qu'il commence par B
- c) Si name possède plus de 5 caractères et qu'il possède plus de 2 caractères ou qu'il commence par B

35. Comment traduire cette condition :

JS

```
if (name && name === "Sohan" || name === "Baptiste" && name.length > 3 && name.length < 10)
```

- a) Si name est défini et qu'il est égal à Sohan ou Baptiste ou qu'il possède entre 3 et 10 caractères.
- b) Si name est défini et qu'il est égal à Sohan et à Baptiste ou qu'il possède entre 3 et 10 caractères.
- c) Si name est défini et qu'il est égal soit à Sohan ou Baptiste et qu'il possède entre 3 et 10 caractères.
- d) Si name est défini et qu'il est égal soit à Sohan ou Baptiste ou qu'il possède entre 3 et 10 caractères.

36. Comment transformer une string en minuscule ?

- a) `.toUpperCase()`
- b) `.ToUpperCase()`
- c) `.toUpper()`
- d) `.toLowerCase()`
- e) `.ToLowerCase()`
- f) `.ToLower`

37. Quels runtimes existent ?

- a) v8
- b) JavaScriptCore
- c) chrome
- d) node

38. Comment vérifier qu'un nombre (**n**) est égal à **NaN** ?

- a) `Number.ISNaN(n)`
- b) `Number.IsNaN(n)`
- c) `Number.isNaN(n)`
- d) `NaN === n`
- e) `NaN == n`
- f) `n.isNaN()`

39. Combien de paramètres prend cette fonction ?

JS

```
const showWarningMessage = (title, message) => {  
  console.log(title)  
  console.log(message)  
}
```

- a) 0
- b) 1
- c) 2
- d) 3

40. Ce code est-il valide ?

```
const showWarningMessage = (title) => { console.log(title) }
```

- a) Oui
- b) Non

41. Quelle méthode est utilisée pour effectuer une requête HTTP avec fetch ? a)

- fetchRequest()
- b) fetch()
- c) getFetch()
- d) httpFetch()

42. Que renvoie la méthode fetch ? a) Un objet XMLHttpRequest

- b) Une promesse
- c) Un tableau
- d) Un objet JSON

43. Quel est le principal avantage de fetch par rapport à XMLHttpRequest ? a) Plus rapide

- b) Basé sur les promesses
- c) Plus de fonctionnalités
- d) Aucune différence

44. Comment gérez-vous les erreurs lors d'une requête fetch ? a) Avec un bloc try...catch
b) Avec .catch()
c) Avec errorHandler()
d) fetch ne peut pas générer d'erreurs

45. Quelle propriété de l'objet XMLHttpRequest vous permet de vérifier l'état de la requête ? a) status
b) readyState
c) responseType
d) responseText

46. Quel type de réponse fetch renvoie par défaut ? a) JSON
b) Texte brut
c) FormData
d) Stream

47. Comment envoyer des données avec fetch ?

- a) En utilisant fetch(url, data)
- b) En utilisant fetch(url, { method: 'POST', body: data })
- c) fetch(url).send(data)
- d) fetch(url).add(data)

48. Quelle méthode XMLHttpRequest est utilisée pour ouvrir une connexion ?

- a) init()
- b) open()
- c) connect()
- d) request()

49. Quelle méthode est utilisée pour obtenir la réponse d'une requête XMLHttpRequest ?

- a) getResponse()
- b) responseText
- c) fetchResponse()
- d) getResponseText()

50. Quelle méthode permet d'annuler une requête XMLHttpRequest en cours ?

- a) abort()
- b) cancel()
- c) stop()
- d) terminate()