

# Créez votre site web avec HTML5 et CSS3

🕒 15 heures 📶 Facile

Mis à jour le 21/12/2022



## Agencer le contenu de pages web

Bravo ! Vous avez réussi cet exercice !

## Compétences évaluées

✅ Agencer le contenu de pages web

### Question 1

Parmi les affirmations suivantes, laquelle est fausse ?

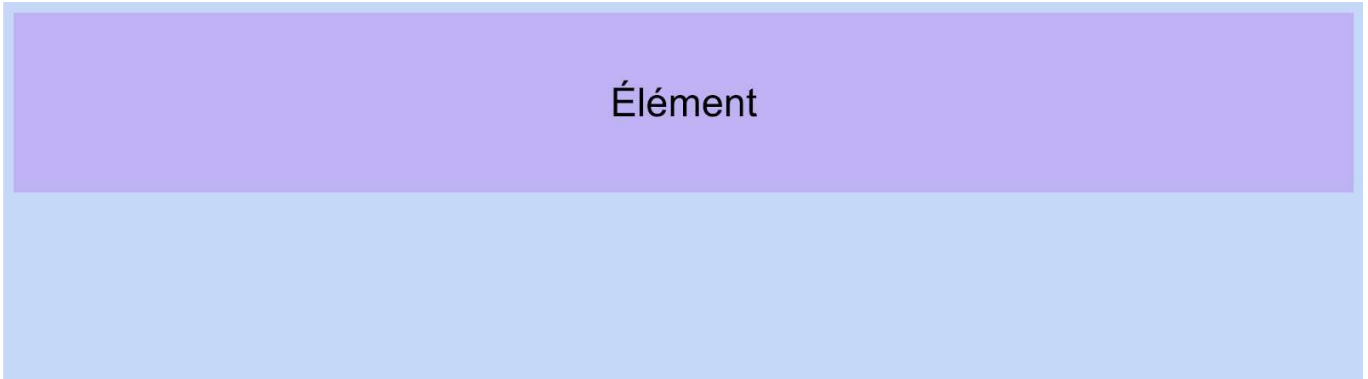
- ☐ Le header sert à indiquer l'en-tête de la page au navigateur
- ☐ On trouve souvent les liens vers les réseaux sociaux et mentions légales dans le footer
- ✅ ☒ L'apparence visuelle du site dépend beaucoup de la structure des balises `<header>` , `<main>` et `<footer>`
- ☐ `<head>` et `<header>` sont deux choses différentes
- ☐ Le contenu principal de la page se trouve dans la balise `<main>`

Les éléments permettant de structurer une page, tels que `<header>` , `<main>` et `<footer>` , ne sont pas indispensables pour l'apparence visuelle d'un site internet. L'affirmation fausse était donc la 3. Cela dit, ils sont utiles pour le navigateur et l'accessibilité : ils indiquent à quelles catégories (en-tête, contenu principal, pied de page...) appartient le contenu.

## Question 2

Regardez le [CodePen Quiz P3Q2](#).

Quel bout de code ajouter dans la classe `.element` pour arriver à la capture d'écran ci-dessous ?



Attention, plusieurs réponses sont possibles.



CSS

```
1 padding: 50px;
```



CSS

```
1 margin: 50px;
```



CSS

```
1 padding-top: 50px;
2 padding-bottom: 50px;
```



CSS

```
1 margin-top: 50px;
2 margin-bottom: 50px;
```

On a besoin de marges intérieures : il faut agir sur les propriétés `padding`. La propriété raccourcie `padding` permet d'ajouter des marges intérieures en haut, à droite, en bas et à gauche. La réponse 1 est donc correcte. Ici, on ne voit que celles en haut et en bas. La réponse 3 marche aussi. En revanche les réponses 2 et 4 ajoutent de la marge : ce n'est pas ce qu'il faut ici.

## Question 3

Parmi les éléments suivants, lesquels sont nativement des éléments `inline` qui vont se mettre les uns à la suite des autres sans prendre toute la largeur ?

Attention, plusieurs réponses sont possibles.

- ☐ main
- ✓ ☒ span
- ✓ ☒ a
- ☐ p

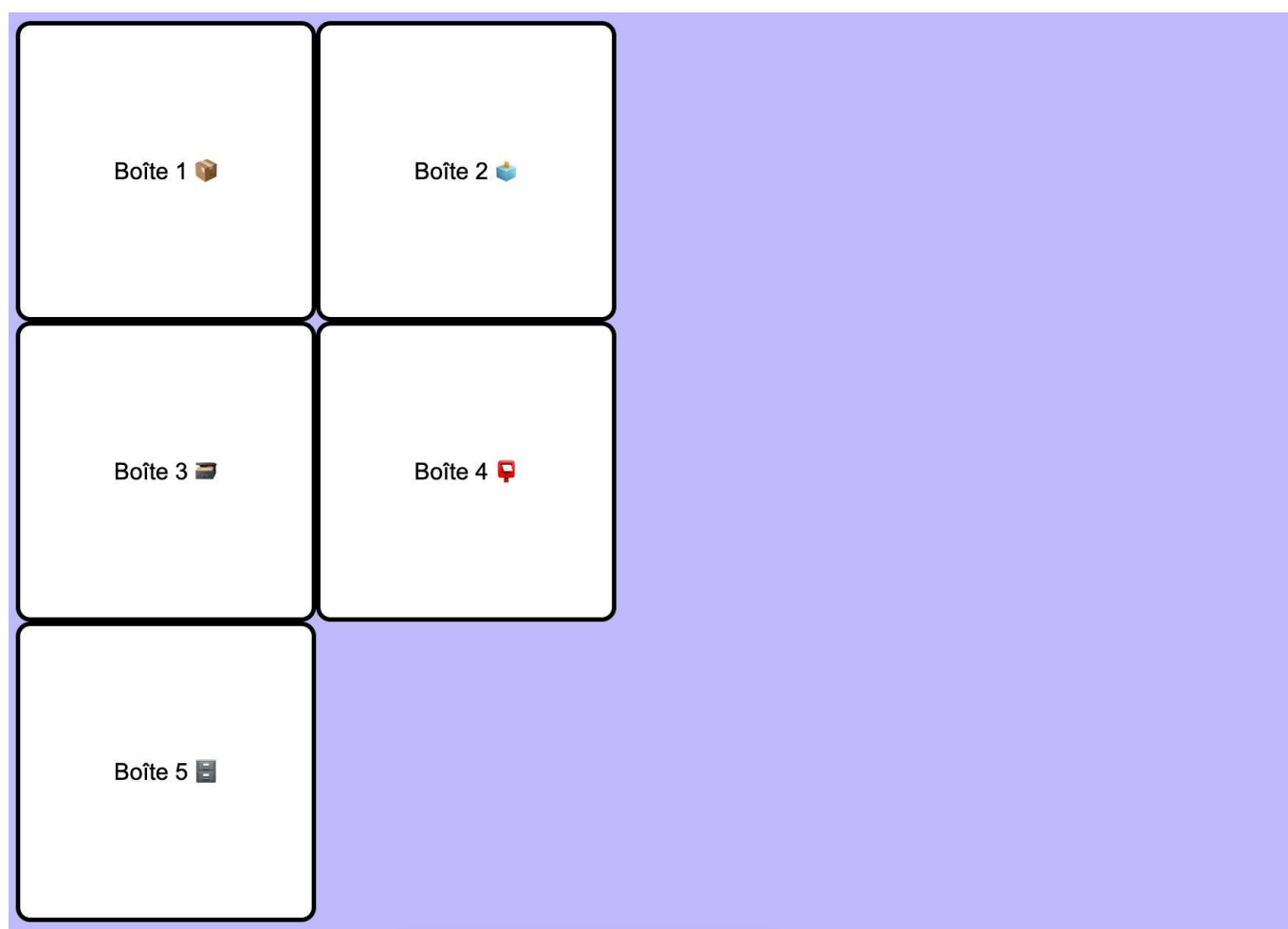
L'élément `main`, ainsi que `p`, tout comme `div` ou les headings, sont des éléments `block`.

En revanche, `span` et `a` sont `inline`. Il fallait donc choisir les réponses 2 et 3.

## Question 4

Regardez le [CodePen Quiz P3Q4](#).

Quelle ligne de code manque-t-il à la classe `.conteneur` pour obtenir le résultat suivant ?



- ✓ ☒ `flex-wrap: wrap;`
- ☐ `justify-content: flex-end;`
- ☐ `margin: auto;`

☐ `display: grid;`

Ici, pas de souci d'alignement : les réponses 2 et 3 ne me sont d'aucune utilité. Par ailleurs, si vous utilisez `display: grid;`, vous verrez que vous obtiendrez une colonne de boîtes, mais pas la même mise en forme. Ici, il fallait bien choisir la réponse 1.

## Question 5

**Quelle propriété appliquée sur un élément dans une mise en page Flexbox permet de modifier l'ordre des éléments en les réagénant sans modifier le HTML ?**

☐ `flex-direction`

☐ `align-items`

✓ ☒ `order`

☐ `flex`

`flex-direction` permet de donner une orientation verticale ou horizontale à vos éléments.

`align-items`, quant à lui, vous permet d'aligner vos éléments sur l'axe secondaire.

Pour ce qui est de `flex`, il s'agit d'une propriété raccourcie qui permet notamment d'autoriser un élément à prendre plus de place que d'autres.

C'est bien `order` qui permet de modifier l'ordre des éléments sans modifier le HTML : réponse 3 !

## Question 6

**Parmi les affirmations suivantes, laquelle est exacte ?**

☐ Les unités `fr` fonctionnent exactement de la même manière que les pourcentages

✓ ☒ Les `gap` permettent d'espacer les éléments entre eux dans les CSS Grids

☐ On définit les colonnes avec `grid-template-rows`

☐ Il est impossible de définir le point de départ d'un élément spécifique dans une `grid`

Les unités `fr` vont permettre de prendre toute la largeur de l'écran sans dépasser, malgré les différents `padding` et `margin`; en cela, elles ne sont pas pareilles que les pourcentages.

Pour leur part, les `gap` vont permettre d'espacer les éléments entre eux de manière simple.

Les colonnes, quant à elles, sont définies par `grid-template-columns` et non `grid-template-rows`.

Enfin, on peut définir le point de départ d'un élément spécifique avec `grid-column-start`. Il fallait donc choisir la réponse 2.

## Question 7

Quelles sont les options offertes par la propriété `display` ?

Attention, plusieurs réponses sont possibles.

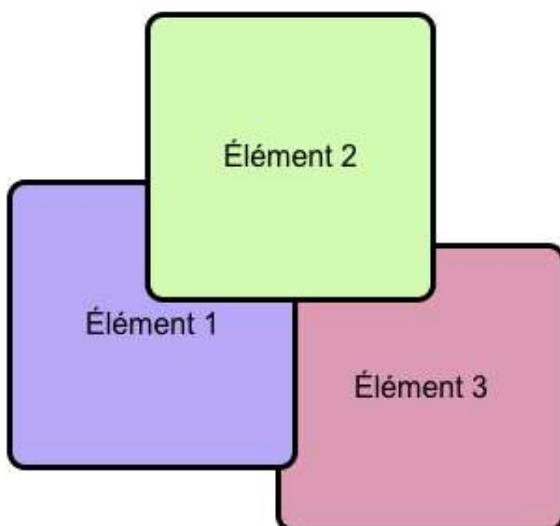
- ✓ ☒ Transformer un élément `inline` en élément `block`
- ✓ ☒ Cacher entièrement un élément
- ✗ ☐ Profiter des avantages de l' `inline` et de ceux des `block`
- ✗ ☒ Superposer des éléments

La propriété `display` est pleine de ressources. Et encore, tout ce qu'elle permet n'est pas couvert par cette question. Elle permet notamment de transformer un élément `inline` en un élément `block`, et inversement. On peut également cacher un élément avec `display: none;`. Et enfin, `display: inline-block` permet de profiter du meilleur des deux mondes : l'élément devient `inline` mais on peut lui assigner une hauteur et une largeur. En revanche, c'est `position` qui permet de superposer des éléments. Il fallait donc choisir les réponses 1, 2 et 3.

## Question 8

Regardez le [CodePen Quiz P3Q8](#).

Quelle propriété permettrait d'obtenir le résultat ci-dessous ?



- ☐ `position`
- ☐ `display`
- ✓ ☒ `z-index`

☐ flex

Nos éléments ont été superposés à l'aide de la propriété `position: absolute;` .

Afin de manipuler l'ordre des superpositions, on peut utiliser la propriété `z-index` . Ici, `flex` et `display` ne nous seront d'aucune utilité. Il fallait donc choisir la réponse 3 !

[⬅ Abordez d'autres techniques de mise en page](#)

[Ajoutez des tableaux ➤](#)

## Les professeurs

### Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

### Alexia Toulmet

Développeuse Fullstack React/Node freelance passionnée par le frontend 🇧🇪

## Découvrez aussi ce cours en...



Livre



PDF

OPENCCLASSROOMS



OPPORTUNITÉS



AIDE



POUR LES ENTREPRISES



EN PLUS



 Français



