

Exercice 1 : Afficher/Cacher un élément en utilisant la classe CSS

Objectif : Manipuler les classes CSS pour afficher ou cacher des éléments.

Enoncé :

1. Créez une page web contenant deux paragraphes `<p>`, l'un visible et l'autre caché via la classe `hidden`.
2. Associez un fichier CSS contenant la règle `.hidden` pour masquer l'élément.
3. Utilisez JavaScript pour retirer la classe `hidden` du deuxième paragraphe au clic d'un bouton.

Code:

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Afficher / Cacher</title>

  <style>
    .hidden {
      display: none;
    }
  </style>
</head>

<body>
  <p id="premier">Texte visible</p>
  <p id="second" class="hidden">Texte caché</p>
  <button id="toggleButton">Afficher le texte caché</button>

  <script>
    document.getElementById('toggleButton').addEventListener('click',
function () {
    var secondParagraph = document.getElementById('second');
    secondParagraph.classList.remove('hidden');
  });
  </script>
</body>

</html>
```

Explication :

- Le paragraphe avec l'id `second` est initialement caché grâce à la classe `hidden` dans le CSS.
- En cliquant sur le bouton, le code JavaScript retire la classe `hidden` du deuxième paragraphe, le rendant visible.

Exercice 2 : Cacher les spoilers par classe

Objectif : Manipuler des éléments en fonction de leurs classes.

Énoncé :

1. Créez une page avec trois paragraphes. Deux d'entre eux auront la classe `spoiler`.
2. Cacher ces spoilers en ajoutant la classe `hidden` au clic d'un bouton.

Code:

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cacher les paras</title>
  <style>
    .hidden {
      display: none;
    }
  </style>
</head>

<body>
  <p>C'est l'histoire d'un garçon troublé</p>
  <p class="para">Car il s'avère que son ours en peluche a disparu</p>
  <p class="para">Mais finalement, il le retrouve</p>
  <button id="hideparas">Cacher les paras</button>

  <script>
    document.getElementById('hideparas').addEventListener('click',
function () {
    var paras = document.getElementsByClassName('para');
    for (var i = 0; i < paras.length; i++) {
      paras[i].classList.add('hidden');
    }
  });
  </script>
</body>

</html>
```

Explication :

- Le bouton "Cacher les paras" va ajouter la classe `hidden` à tous les éléments ayant la classe `para`, ce qui les rend invisibles.

Exercice 3 : Surbrillance au clic d'un paragraphe

Objectif : Modifier dynamiquement le style CSS d'un élément.

Énoncé :

1. Créez une page avec deux paragraphes.
2. Ajoutez une interaction où, au clic sur le deuxième paragraphe, sa couleur de fond devient jaune.

Code:

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Surbrillance au clic</title>
</head>

<body>
  <p id="premier">Texte important</p>
  <p id="second">Autre texte</p>

  <script>
    document.getElementById('second').addEventListener('click',
function () {
    this.style.backgroundColor = 'yellow';
    });

    document.getElementById('premier').addEventListener('click',
function () {
    this.style.backgroundColor = 'red';
    });

  </script>
</body>

</html>
```

Explication :

- Lorsqu'on clique sur le deuxième paragraphe, sa couleur de fond devient jaune grâce à la modification de son style via JavaScript.

Exercice 4 :

Créez une page HTML avec trois paragraphes et deux boutons.

- Utilisez **querySelectorAll** pour sélectionner tous les paragraphes.
- Changez la couleur de fond des paragraphes en cliquant sur un bouton.

Correction :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Exercice querySelectorAll</title>
</head>
<body>
  <p>Paragraphe 1</p>
```

```
<p>Paragraphe 2</p>
<p>Paragraphe 3</p>
<button id="colorButton">Changer la couleur</button>

<script>
    document.getElementById('colorButton').addEventListener('click',
function () {
    document.querySelectorAll('p').forEach(function (para) {
        para.style.backgroundColor = 'lightblue';
    });
});
</script>
</body>
</html>
```

5. Exercices pour `getAttribute`

Objectif : Apprendre à lire les attributs d'un élément HTML.

Créez une liste d'images. Lorsque vous cliquez sur une image, affichez son attribut `src` dans la console.

Correction :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Exercice getAttribute</title>
</head>
<body>
    
    
    

    <script>
        document.querySelectorAll('img').forEach(function (image) {
            image.addEventListener('click', function () {
                console.log(this.getAttribute('src'));
            });
        });
    </script>
</body>
</html>
```

6. Exercices pour `setAttribute`

Objectif : Apprendre à modifier ou ajouter des attributs à des éléments.

Créez une liste de trois liens. Ajoutez un bouton qui, lorsqu'il est cliqué, change l'attribut `href` des liens pour pointer vers `https://www.google.com`.

Correction :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Exercice setAttribute</title>
</head>
<body>
  <a href="#">Lien 1</a>
  <a href="#">Lien 2</a>
  <a href="#">Lien 3</a>
  <button id="changeLinks">Changer les liens</button>

  <script>
    document.getElementById('changeLinks').addEventListener('click',
function () {
    document.querySelectorAll('a').forEach(function (link) {
      link.setAttribute('href', 'https://www.google.com');
    });
  });
</script>
</body>
</html>
```

7. Exercices pour contains

Objectif : Vérifier si une classe est présente sur un élément.

Créez une liste d'articles avec des classes comme `important` et `highlight`. Ajoutez un bouton qui, lorsqu'il est cliqué, affiche dans la console les articles qui ont la classe `important`.

Correction :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Exercice contains</title>
</head>
<body>
  <article class="important">Article 1</article>
  <article class="highlight">Article 2</article>
  <article class="important highlight">Article 3</article>
  <article>Article 4</article>
  <button id="checkImportant">Afficher les importants</button>

  <script>
    document.getElementById('checkImportant').addEventListener('click',
function () {
    document.querySelectorAll('article').forEach(function (article)
{
      if (article.classList.contains('important')) {
        console.log(article.textContent);
      }
    });
  });
</script>
</body>
```

```
</html>
```

Exercice 8 : Filtrage dynamique de produits par catégorie

Objectif : Créer une liste de produits filtrable par catégories.

Énoncé :

1. Créez une page avec une liste de produits et des boutons permettant de filtrer les produits par catégorie.
2. Vous devez inclure au moins 10 produits et 3 catégories.
3. Utilisez les classes pour associer chaque produit aux catégories et filtrez les produits en fonction des catégories sélectionnées.

Code HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Filtrage par catégorie</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <button class="filter" data-category="ludique">Ludique</button>
  <button class="filter" data-category="pratique">Pratique</button>
  <button class="filter" data-category="sportif">Sportif</button>
  <button class="filter" data-category="all">Tous</button>

  <article class="ludique pratique">iPad</article>
  <article class="ludique sportif">Batte de baseball</article>
  <article class="pratique">Ventilateur</article>
  <article class="sportif">Tennis</article>
  <article class="ludique">Jeu vidéo</article>
  <article class="pratique">Sac à dos</article>
  <article class="ludique sportif">Skateboard</article>
  <article class="pratique">Montre connectée</article>
  <article class="ludique">Puzzle</article>
  <article class="sportif">Vélo</article>

  <script src="script.js"></script>
</body>
</html>
```

Code JavaScript (script.js) :

```
document.querySelectorAll('.filter').forEach(function(button) {
  button.addEventListener('click', function() {
    var category = this.getAttribute('data-category');
    var products = document.querySelectorAll('article');

    products.forEach(function(product) {
```

```
        if (category === 'all' || product.classList.contains(category))
        {
            product.style.display = 'block';
        } else {
            product.style.display = 'none';
        }
    });
});
});
```

Explication :

- Chaque bouton de filtrage correspond à une catégorie. Quand un bouton est cliqué, le code JavaScript affiche uniquement les produits qui appartiennent à la catégorie sélectionnée. Si l'utilisateur clique sur "Tous", tous les produits sont affichés.