

Développement web S1 (R112) – TP #6

Liens utiles : [CM #1](#) · [TP #2](#) · [TP #3](#) · [TP #4](#) · [CM #2](#)

Avant de commencer, créez un répertoire nommé r112-tp6 contenant des répertoires exo1, exo2, etc. Chacun de ces répertoires contient votre travail sur l'exercice correspondant. Le fichier HTML de chaque exercice doit s'appeler index.html. **Vos exercices ne seront pas corrigés si vous ne respectez pas ces instructions.**

À la fin de la séance, vous devez compresser (en format ZIP ou RAR) le répertoire r112-tp6 avec tous vos exercices et le soumettre sur AMeTICE.

Seuls les exercices marqués avec une étoile (*) seront évalués. La date limite pour rendre vos exercices est visible sur AMeTICE, et aucun travail ne sera accepté après. Le corrigé sera accessible après l'évaluation des exercices.

Exercices

Exercice 1

Dans cet exercice vous allez modifier tous les liens d'une page web.

Pour chaque lien ([a](#)), vous allez ajouter avec JavaScript l'attribut `target="_blank"` afin que la page soit ouverte dans un nouvel onglet.

Utilisez le code HTML suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>R112</title>
</head>
<body>
  <h1>Jethro Tull</h1>
  <p>
    Les albums studios de Jethro Tull sont :
  </p>
  <ul>
    <li><a href="https://fr.wikipedia.org/wiki/This_Was">This Was</a> (1968)</li>
    <li><a href="https://fr.wikipedia.org/wiki/Stand_Up_(album_de_Jethro_Tull)">Stand Up</a> (1969)</li>
  </ul>
</body>
</html>
```

Utiliser des liens qui s'ouvrent sur un nouvel onglet évite que les visiteurs de votre site web l'abandonnent durant la navigation. Attention à ne pas abuser de cette technique.

On peut définir ou modifier l'attribut d'un élément HTML avec la méthode `setAttribute()` [\[doc\]](#).

Exercice 2

Dans cet exercice vous allez faire un bouton qui affiche un contenu en utilisant une classe.

Commencez par copier le code HTML suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>R112</title>
  <style>
    .cache {
      display: none;
    }
    #motdepasse {
      color: red;
    }
  </style>
</head>
<body>
  <div id="motdepasse">
    <span class="cache">
      Le mot de passe est :
    </span>
  </div>
</body>
</html>
```

Quand l'utilisateur clique sur le bouton, il faut retirer la classe `cache` de l'élément `` pour que le mot de passe soit visible.

On peut ajouter et retirer des classes d'un élément HTML avec les méthodes `classList.add()` et `classList.remove()`. Par exemple :

```
let h1 = document.querySelector("h1") // <h1 class="centre">
h1.classList.add("titre-principal")    // <h1 class="centre titre-principal">
h1.classList.remove("centre")         // <h1 class="titre-principal">
```

Exercice 3

Dans cet exercice vous allez faire un bouton qui affiche ou cache un contenu avec un bouton.

Reprenez votre code pour l'exercice précédent et faites les modifications suivantes :

- Quand l'utilisateur clique sur le bouton, le mot de passe est affiché. Le texte du bouton devient alors «Cacher»
- Quand on clique sur le bouton «Cacher», le mot de passe disparaît et le texte du bouton devient «Afficher»

Vous pouvez savoir si une classe apparaît dans un élément HTML avec la méthode `classList.contains()` :

```
let h1 = document.querySelector("h1") // <h1 class="centre">
h1.classList.contains("titre-principal") // false
h1.classList.contains("centre")         // true
```

Exercice 4* noté

Dans cet exercice vous allez faire un accordéon : les contenus de la page sont affichés ou cachés avec des boutons.

Copiez le code HTML suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>R112</title>
  <style>
    .cache {
      display: none;
    }
  </style>
</head>
<body>
  <h1>Films de Quentin Tarantino</h1>
  <div id="films">
    <div class="cache">
      <div id="film1">
        <h2>Pulp Fiction</h2>
        <p>Un couple de policiers se retrouve dans une histoire d'amour...</p>
      </div>
      <div id="film2">
        <h2>Kill Bill</h2>
        <p>Une femme se venge d'un mari qui a tué sa famille...</p>
      </div>
      <div id="film3">
        <h2>Inglourious Basterds</h2>
        <p>Des soldats américains se battent contre les nazis...</p>
      </div>
    </div>
  </div>
</body>
</html>
```

Dans cette page, les descriptions des films sont cachées par défaut. Comme dans l'exercice précédent, ajoutez des événements dans cette page pour que chaque bouton affiche ou cache la description de son film.

La difficulté de cet exercice est qu'il faut trouver le paragraphe à partir du bouton. En JavaScript il existe deux méthodes pour descendre et monter dans l'arborescence du DOM :

- La méthode `querySelector()` d'un élément HTML prend un paramètre un sélecteur CSS (comme `document.querySelector()`) et renvoie le premier élément HTML dans son intérieur qui matche avec le sélecteur CSS.
- La méthode `closest()` fait le contraire. Elle remonte dans l'arborescence et renvoie le premier élément HTML qui matche avec le sélecteur CSS.

Exercice 5

Dans cet exercice vous allez filtrer les éléments d'une liste à l'aide des boutons.

On vous donne le code HTML et JavaScript suivant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>R112</title>
</head>
<body>
  <h1>Étudiants</h1>
  <p>
    <button id="promo">Promotion</button>
    <button id="tda">Groupe TDA</button>
    <button id="tdb">Groupe TDB</button>
    <button id="tp1">Groupe TP1</button>
    <button id="tp2">Groupe TP2</button>
  </p>
  <div id="etudiants">
    <div id="etudiants1">
      <div id="etudiants1_1">Adèle MICHELIER</div>
      <div id="etudiants1_2">Alexandre LACOTE</div>
      <div id="etudiants1_3">Alexia FOUCHARD</div>
      <div id="etudiants1_4">Arthur LAMPIN</div>
      <div id="etudiants1_5">Baptiste MORET</div>
      <div id="etudiants1_6">Benjamin ONA</div>
      <div id="etudiants1_7">Camille ROGER</div>
      <div id="etudiants1_8">Caroline CLAUDEL</div>
      <div id="etudiants1_9">Caroline BION</div>
      <div id="etudiants1_10">Claire LAZERAT</div>
      <div id="etudiants1_11">Clémentine RIGUET</div>
      <div id="etudiants1_12">Damien MOURGUES</div>
      <div id="etudiants1_13">Diego GROSNI</div>
    </div>
  </div>
</body>
</html>
```

```
let etudiants = [
  { "nom": "Adèle MICHELIER", "groupe": "tp1" },
  { "nom": "Alexandre LACOTE", "groupe": "tp1" },
  { "nom": "Alexia FOUCHARD", "groupe": "tp1" },
  { "nom": "Arthur LAMPIN", "groupe": "tp1" },
  { "nom": "Baptiste MORET", "groupe": "tp1" },
  { "nom": "Benjamin ONA", "groupe": "tp1" },
  { "nom": "Camille ROGER", "groupe": "tp1" },
  { "nom": "Caroline CLAUDEL", "groupe": "tp1" },
  { "nom": "Caroline BION", "groupe": "tp1" },
  { "nom": "Claire LAZERAT", "groupe": "tp1" },
  { "nom": "Clémentine RIGUET", "groupe": "tp1" },
  { "nom": "Damien MOURGUES", "groupe": "tp2" },
  { "nom": "Diego GROSNI", "groupe": "tp2" },
]
```

À chaque fois que l'utilisateur clique sur un bouton, il faut afficher les étudiants correspondants dans la liste non ordonnée (``).

Exercice 6

Refaites l'[exercice 5](#), mais cette fois en utilisant la même fonction pour tous les boutons.