

TP N°1 partie 2 : découverte de JQuery

Pourquoi les sélecteurs sont les éléments les plus importants de jquery

Les sélecteurs sont les éléments les plus importants de jQuery car ils permettent d'interagir avec les éléments du DOM (Document Object Model) d'une page web.

Les sélecteurs jouent un rôle central dans l'utilisation de jQuery pour les raisons suivantes :

1. **Sélection d'éléments** : Les sélecteurs permettent d'identifier et de sélectionner les éléments du DOM avec lesquels vous souhaitez travailler. Vous pouvez sélectionner des éléments par ID, classe, nom de balise, attributs, etc.
2. **Manipulation du DOM** : Une fois que vous avez sélectionné un élément ou un ensemble d'éléments, vous pouvez utiliser jQuery pour manipuler ces éléments, en modifiant leur contenu, leurs attributs, leurs styles, etc.
3. **Gestion des événements** : Les sélecteurs sont également utilisés pour attacher des gestionnaires d'événements aux éléments sélectionnés, permettant d'écouter et de réagir aux événements (par exemple, les clics de souris, les appuis sur des touches du clavier, etc.) sur les éléments du DOM.
4. **Facilité d'utilisation** : Les sélecteurs jQuery sont basés sur les sélecteurs CSS, ce qui les rend faciles à comprendre et à utiliser pour les développeurs ayant une expérience en CSS. De plus, jQuery offre une syntaxe concise et simple pour travailler avec les éléments du DOM.

I. Première page JQuery

1. La page Web

Voici un exemple simple de projet qui utilise HTML, CSS, JavaScript et jQuery pour créer un bouton qui, lorsqu'on clique dessus, ajoute un élément de liste à une liste non ordonnée.

a) Le CSS est :

```
body {  
  font-family: Arial, sans-serif;  
  text-align: center;  
}  
  
button {
```

```
background-color: #4CAF50;
color: white;
padding: 10px 20px;
border: none;
cursor: pointer;
font-size: 16px;
margin: 10px;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  background-color: #f1f1f1;
  padding: 8px 16px;
  margin: 5px;
}
```

b) Et le javascript/JQuery est

```
$(document).ready(function() {
  $("body").append("<h1>Hello World avec jQuery</h1>");
});
```

c) Le fichier HTML

Question 1. Complétez la balise head de ce fichier pour pouvoir utiliser JQuery (la version jquery-3.6.4.min.js)

```
<!DOCTYPE html>
<html lang="fr">
<head>
...
</head>
<body>
  <h1>Liste de tâches</h1>
  <button id="ajouter">Ajouter une tâche</button>
  <ul id="listeTaches"></ul>
</body>
</html>
```

Vous devez obtenir cet affichage :

Liste de tâches

Ajouter une tâche

Hello World avec jQuery

Question 2. Expliquez les lignes ci-dessous :

```
$(document).ready(function() {  
    $("body").append("<h1>Hello World avec jQuery</h1>");  
});
```

Maintenant nous allons vouloir mettre une fonction anonyme callback sur l'appui bouton. Ainsi à chaque fois qu'on appuie sur *Ajout Tâche* du texte est ajoutée de cette façon après 4 appuis :

Liste de tâches

Ajouter une tâche

Tâche 1

Tâche 2

Tâche 3

Tâche 4

Question 3. Complétez le fichier JS/JQuery afin d'obtenir ce comportement :

```
$(document).ready(function() {
  $("body").append("<h1>Hello World avec jQuery</h1>");

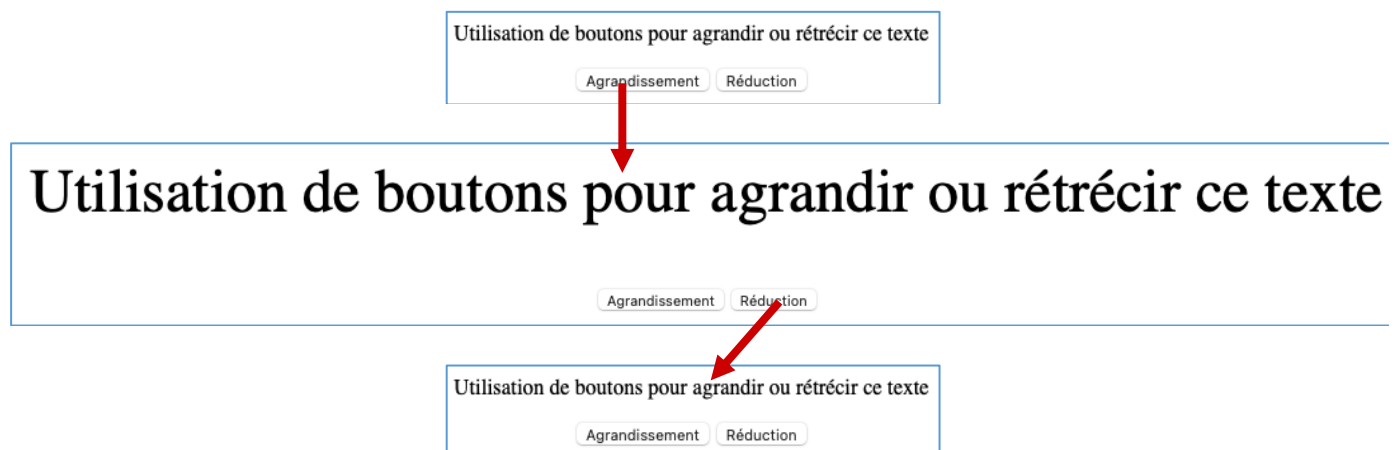
  let compteur = 1;

  $("#ajouter").click(function() {
    $("#_____").append('<li>Tâche ' + _____ + '</li>');
    Compteur_____;
  });
});
```

II. Jouons avec les boutons

1. Présentation

Dans cette partie vous allez mettre en place une page web avec deux boutons : un qui agrandit le texte du paragraphe et l'autre qui le réduit à sa taille initiale :



2. HTML

Question 4. Écrivez une page HTML où vous avez une balise `<div>` qui englobe plusieurs éléments, un paragraphe `<p>` et deux boutons `<button>` :

- Un premier paragraphe qui affiche **Utilisation de boutons pour agrandir ou rétrécir ce texte**
- Deux boutons, ayant respectivement les identifiants **agrandissement** et **reduction**

3. Fichier CSS

Question 5. Écrivez un css qui respecte ces règles :

1. Une règle définit la taille de la police pour tous les éléments `<p>` (paragraphe). La taille de la police est définie sur **1em** (**1em** signifie que la taille de la police des paragraphes sera la même que celle héritée de l'élément parent).

- Une autre règle qui définit la taille de la police pour les éléments `<p>` ayant la classe **grand**. La taille de la police est définie sur **3em**.
- Une dernière règle qui aligne le texte au milieu pour tous les `<DIV>`

4. JavaScript/ JQuery

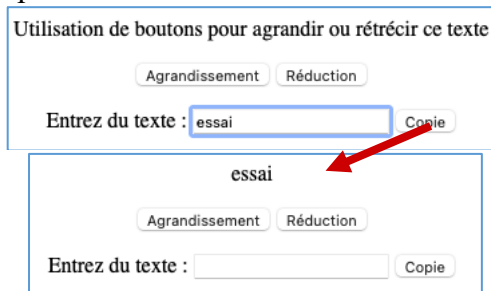
La méthode `addClass` ajoute une classe à un élément tandis que `removeClass` en supprime (c.f. https://www.w3schools.com/jquery/html_removeclass.asp)

Question 6. Écrivez le code jQuery dans lequel est défini deux gestionnaires de clic pour les boutons "réduction" et "agrandissement" qui modifient la taille du texte des éléments `<p>` en ajoutant ou supprimant la classe "grand".

5. Nouveau bouton avec un label

a) But

Le but de cette partie est d'ajouter une étiquette, un input (champ de texte) et un bouton qui va remplacer le texte du paragraphe par celui de l'input :



b) L'HTML

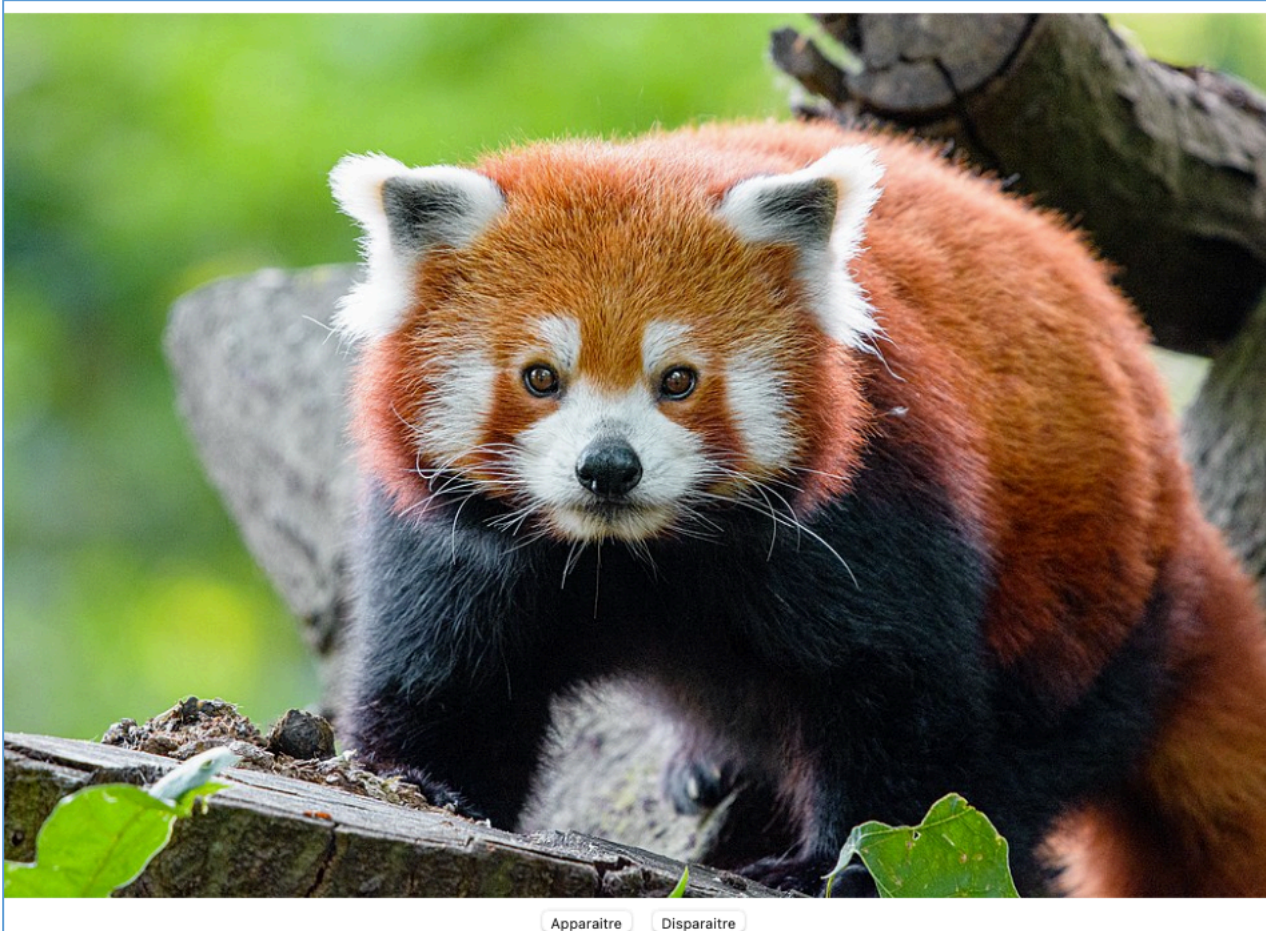
Question 7. Ajoutez à votre page HTML un paragraphe `<p>` contenant un label et un bouton *Copie*. Ne pas utiliser d'identifiant ni pour le bouton ni pour le label

c) JQuery

Dans cette partie, il est interdit d'utiliser les identifiants, vous devez utiliser les sélecteurs sur les balises, avec si nécessaire la méthode `eq(index)` avec l'index approprié. Par ailleurs, en JQuery, la valeur contenu dans un input se récupère non pas avec l'attribut `.value` comme en javascript mais avec la méthode `.val()`, méthode qui permet également de modifier cet attribut.

Question 8. Écrivez le code jQuery qui sélectionne le troisième bouton de la page et lui associe un gestionnaire de clic. Lorsque ce bouton est cliqué, il copie la valeur de l'élément `<input>` dans le premier paragraphe (`<p>`) et vide ensuite la valeur de l'élément `<input>`.

III. FadeIn FadeOut



1. Rappel

En jQuery, les méthodes **fadeIn()** et **fadeOut()** sont utilisées pour animer progressivement la transparence d'un élément.

- **fadeIn()**: Cette méthode anime l'élément en augmentant progressivement sa transparence, passant de complètement transparent (opacity 0) à complètement opaque (opacity 1). En d'autres termes, elle fait apparaître l'élément sur la page. Vous pouvez également spécifier une durée pour l'animation en millisecondes en tant que paramètre, par exemple **fadeIn(500)** pour une animation de 500 millisecondes.
- **fadeOut()**: Cette méthode fait l'inverse de **fadeIn()**. Comme pour **fadeIn()**, vous pouvez spécifier une durée pour l'animation en millisecondes en tant que paramètre.

2. Le CSS

Question 9. Écrivez le CSS en suivant ces consignes :

Ce CSS permet de centrer le contenu de la page, d'organiser les éléments enfants du **.container** en colonne et de gérer l'espacement des boutons.

Pour cela :

1. **body** :
 - **display: flex;** Utilise le modèle de boîte flexible (Flexbox) pour disposer les éléments enfants du **body**.
 - **justify-content: center;** Centre horizontalement les éléments enfants du **body**.
 - **align-items: center;** Centre verticalement les éléments enfants du **body**.
 - **min-height: 100vh;** Définit une hauteur minimale pour le **body** égale à 100% de la hauteur de la fenêtre de visualisation (viewport height).
 - **margin: 0;** Supprime les marges par défaut autour du **body**.
 - **font-family: Arial, sans-serif;** Utilise la police Arial comme police principale et une police sans-serif comme police de secours si Arial n'est pas disponible.
2. **.container** :
 - **display: flex;** Utilise le modèle de boîte flexible (Flexbox) pour disposer les éléments enfants du **.container**.
 - **flex-direction: column;** Modifie l'orientation des éléments enfants du **.container** pour les disposer en colonne plutôt qu'en ligne (par défaut).
 - **align-items: center;** Centre horizontalement les éléments enfants du **.container**.
3. **button** :
 - **margin-top: 10px;** Ajoute une marge de 10px au-dessus de chaque bouton.
 - **margin-right: 10px;** Ajoute une marge de 10px à droite de chaque bouton.
4. **button:last-child** :
 - **margin-right: 0;** Supprime la marge à droite du dernier bouton. Cette règle cible spécifiquement le dernier élément **button** parmi les enfants de son parent, afin qu'il n'y ait pas de marge à droite inutile pour le dernier bouton.

3. L'HTML

Question 10. L'HTML contient avec une `<div>` ayant la classe "container". Cette `<div>` contient :

- Une balise `` qui affiche une image dont le fichier source est une image sur votre ordinateur (ou https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/The_red_panda_%28Ailurus_fulgens%29_1.jpg/1024px-The_red_panda_%28Ailurus_fulgens%29_1.jpg)
- Une autre `<div>` qui englobe deux boutons :
- Un bouton avec le texte "Bouton 1".
- Un second bouton avec le texte "Bouton 2".
- Pas d'utilisation d'identifiant

4. JQuery

Question 11. Mettez un gestionnaire d'événements de click sur les deux boutons, un va masquer l'image et le second va l'afficher avec en une seconde.