

# **Animations Regal'recup avec du JQuery**

## Liste des animations ***Jquery*** de la page Régal'récup

- 1/Animation du titre H1**
- 2/Amimation de la flèche**
- 3/Animation des titres H2**
- 4/Transtion d'image**
- 5/Paralaxe au scroll**
- 6/Apparition d'image**

# Explication du choix du **HTML CSS JQUERY**

images	28/02/2025 21:26	Dossier de fichiers	
Heavitas	25/02/2025 23:42	Fichier de police T...	120 Ko
index	28/02/2025 22:39	Fichier HTML	5 Ko
index	28/02/2025 22:36	Fichier source Jav...	4 Ko
indexjs	28/02/2025 22:39	Fichier source Jav...	4 Ko
jquery	28/02/2025 11:49	Fichier source Jav...	85 Ko
Roboto-Regular	25/02/2025 23:42	Fichier de police T...	155 Ko
styles	28/02/2025 22:29	Fichier source CSS	8 Ko

Le dossier images contient toutes les images qui vont être utilisés sur le site.

J'ai réduit et supprimé les fichiers du dossier au maximum pour faciliter ma zone de travail.  
Le code, n'étant pas très bien organisé, cela a facilité la création du code et des animations prévues.

## Choix:

J'ai décidé de travailler sur le dossier **Régál'recup2** sur une page js grâce à JQuery

La réduction de la zone de travail m'a permis de mieux identifier les zones où je devais retravailler dans un code qui n'était pas le mien.

<https://jqueryui.com/> Les bibliothèques JQuery ont été prise dessus

J'ai supposé le le fichier **jquery.js** était indispensable au bon fonctionnement du code je l'ai donc laissé tel quel .

```
<script type="text/javascript" src="jquery.js"></script>
```

J'ai intégré le fichier **index.js** dans lequel il y aura les éléments du site animés grâce a du jquery et le css modifié.

```
<script type="text/javascript" src="index.js"></script>
```

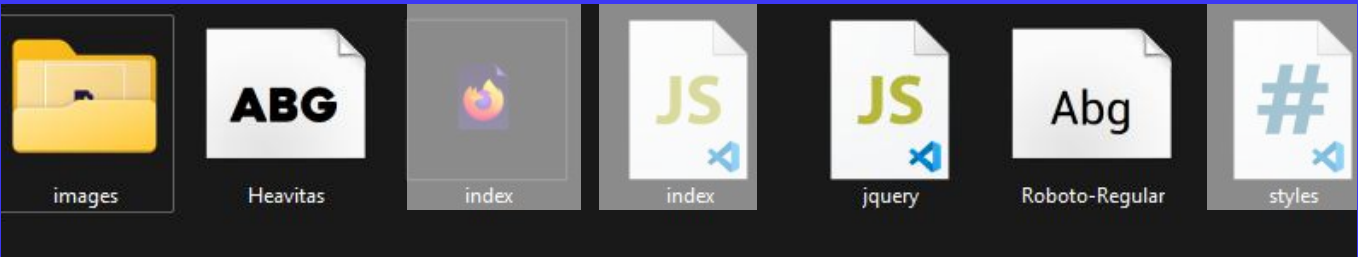
J'ai laissé le fichier **styles.css** dans lequel il y a les éléments de style un html qui je vais modifié par la suite.

```
<link type="text/css" href="styles.css" rel="stylesheet">
```

J'ai aussi importé la bibilothèque JQuery UI qui permet l'ajout d'animation interactives.

```
<script src="https://code.jquery.com/ui/1.13.1/jquery-ui.min.js" integrity="sha256-eTyxS0rkjpLEo16uXTS0uVCS4815lc40K2iVpWDvdSY=" crossorigin="anonymous"></script>
```

```
<link type="text/css" href="styles.css" rel="stylesheet">
<script type="text/javascript" src="index.js"></script>
<script type="text/javascript" src="jquery.js"></script>
<script src="https://code.jquery.com/ui/1.13.1/jquery-ui.min.js" integrity="sha256-eTyxS0rkjpLEo16uXTS0uVCS4815lc40K2iVpWDvdSY=" crossorigin="anonymous"></script>
```





# Animation 1:

Exemple du code du professeur mit dans un éditeur de code.(Vs code)

```
const word = document.querySelector('.word');
const letters = word.textContent.split('');
word.textContent = '';

letters.forEach(letter => {
  const span = document.createElement('span');
  span.textContent = letter;
  span.style.transition = 'all 0.3s';
  word.appendChild(span);
});

word.addEventListener('mouseover', () => {
  word.querySelectorAll('span').forEach(letter => {
    letter.style.transform = 'translateX(20px) scale(1.5)';
  });
});

word.addEventListener('mouseout', () => {
  word.querySelectorAll('span').forEach(letter => {
    letter.style.transform = 'translateX(0) scale(1)';
  });
});
```



```
// Effet au survol
$(document).ready('mouseover', () => {
  titre.querySelectorAll('span').forEach(letter => {
    letter.style.transform = 'translateY(0px) scale(1) rotateY(180deg)';
  });
});

$(document).ready('mouseout', () => {
  titre.querySelectorAll('span').forEach(letter => {
    letter.style.opacity = '1';
    letter.style.transform = 'translateY(0px) scale(1) rotateY(360deg)';
    letter.style.color = 'red';
  });
});
}
```

**1/** Lorsque la souris survole l'élément titre, chaque lettre dans l'élément **<span>** est affectée.

La transformation de chaque lettre est modifiée pour **TranslateY(0px)** on le remet la la position verticale,  
On met aussi **Scale(1)** pour la taille normale de la lettre  
Et **RotateY(180deg)** : Rotation à 180 degrés sur l'axe Y, ce qui inverse l'effet de rotation initial

**2/** Lorsque la souris quitte l'élément titre avec **mouseout**, chaque lettre retourne à son état initial.

L'opacité reste 1 avec **opacity = 1**  
La transformation des lettres est réinitialisée avec **rotateY(360deg)**, ce qui les fait revenir à leur rotation initiale.  
La couleur des lettres est changée en vert avec **letter.style.color = 'green'**

# Titre animé

J'ai décidé se faire arriver les lettres une par une lors du chargement de la page.  
Cela concerne les lettres du texte dans **#titre1** qui apparaissent progressivement avec une **animation de rotation**.

J'ai aussi fais un effet de survol : Lors du survol de la souris sur le texte, chaque lettre effectue **une transformation (rotation de 180°)**.

Pour une meilleur fluidité j'ai rajouté un retour à l'état inital lorsque la souris quitte **le texte**, chaque lettre revient à son état initial (avec une **rotation 360°**) et **devient verte**.

## Animation 2:

```
// Animation de la flèche
function fleche() {
  $('#fleche')
    .animate({ top: '-=20px' }, 300)
    .animate({ top: '+=20px' }, 300)
    .animate({ rotate: '+=10deg' }, 200)
    .animate({ rotate: '-=10deg' }, 200)
    .animate({ rotate: '-=10deg' }, 200)
    .animate({ rotate: '+=10deg' }, 200, fleche);

  // Démarre l'animation de la flèche
  fleche();
}
```

→ Cette partie du code lance la fonction fleche()

→ **1/** Déplace l'élément vers le haut de **20 pixels** avec top et “-= ” avec une durée de **300 ms** et  
**2/** inversement pour “+= ” qui amène la flèche vers le bas.

→ **3/** Fait tourner l'élément de **10 degrés** dans le sens horaire grâce à rotate: **'+=10deg'** avec une durée de **200 ms** puis  
**4/** inversement avec une rotation dans le sens antihoraire grâce à rotate: **'-=10deg'** avec une durée de **200 ms**.  
**5/** Ces deux animations sont répétés deux fois.

→ Cette partie du code lance la fonction fleche() en boucle



## Flèche:

L'élément avec l'ID **#fleche** se déplace vers le haut et vers le bas en continu et effectue une série de rotations à gauche et à droite.

Ces mouvements se produisent de manière fluide et continue grâce à la **boucle fleche** qui rappelle la fonction après chaque cycle.

Cette effet visuel attirer l'attention sur la flèche qui indique la direction du regard.

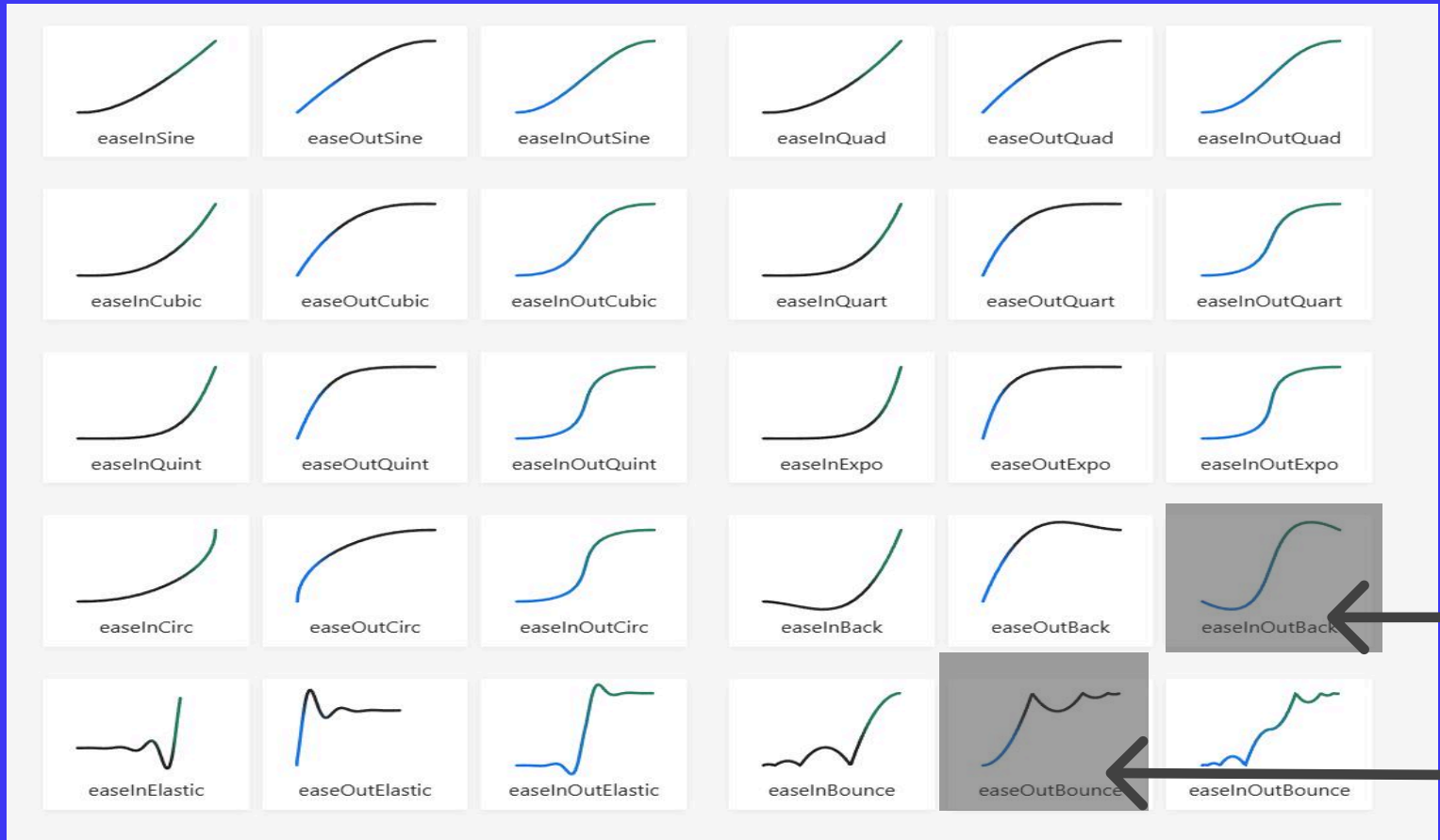


# Animation 3:

## Animation des titres h2

```
//Animation de tous les titres h2
$(document).ready(function(){
  $('h2').css({ position: 'relative', top: '-400px' }).animate({ top: '15px' }, 800, 'easeOutBounce');

  $('h2').on('mouseover', function() {
    $(this).animate({ color: 'red' }, 600, 'easeInOutBack');
  });
});
```



1/ Les titres **<h2>** sont initialement placés hors de la vue à **top: -400px** et se déplacent vers leur position finale **top: 15px** avec une animation fluide de **800ms**. L'animation utilise l'effet **easeOutBounce**, ce qui crée un effet de rebond à la fin du mouvement.

2/ Lorsqu'un titre **<h2>** est survolé, sa couleur change en rouge grâce à une animation de **600ms** avec l'effet **easeInOutBack**. L'animation commence lentement, accélère, puis ralentit avant d'atteindre la couleur finale.

[Easings.net](https://easings.net)

**PS: Relancer la page pour les aniamtions des titres h2**



## Titres en h2

Les titres **<h2>** arrivent sur la page de haut en bas avec un effet de rebond.

Lorsqu'un titre **<h2>** est survolé, sa couleur change en **rouge** avec une animation fluide.

Les animations des titres ont été inspirées par le site [easings.net](https://easings.net)

## Animation 4:

### Transitions d'images:

```
$(document).ready(function() {
  $('.ptips .imgtips').on('mouseenter', function() {
    $(this).fadeOut(500);
    $(this).siblings('.imgtips1').delay(500).fadeIn(500);
  });
  $('.ptips .imgtips1').on('mouseleave', function() {
    $(this).fadeOut(500);
    $(this).siblings('.imgtips').delay(500).fadeIn(500);
  });
});
```

**1/** L'image avec la classe **.imgtips** disparaît progressivement en **500 ms** grâce à un **fadeOut()**.

Après un délai de **500 ms**, l'autre image avec la classe **.imgtips1**, sélectionnée grâce à **.siblings()**, apparaît progressivement en **500 ms** grâce à **fadeIn()**.

**2/** L'image avec la classe **.imgtips1** disparaît progressivement en 500 ms grâce à **fadeOut()**.

Après un délai de 500 ms (delay(500)), l'autre image avec la classe **.imgtips** apparaît progressivement en **500 ms** grâce à **fadeIn()**.

### Transitions d'images:

Au survol de l'image **.imgtips**, celle-ci disparaît, et l'image **.imgtips1** du même bloc apparaît.

Quand la souris ne se trouve plus sur l'image **.imgtips1**, celle-ci disparaît, et l'image **.imgtips** du même bloc réapparaît.

J'avais besoin de sélectionner 2 images pour mon animation, j'ai finalement trouvé l'emploi de **siblings()**.

Le site <https://api.jquery.com/siblings/> m'a aidé à faire l'animation prévu.





# Animation 5:

## Effet de parallaxe au scroll

```
$(document).ready(function () {
  const img1 = document.getElementById('imgb1');
  const img3 = document.getElementById('imgb3');
  const img4 = document.getElementById('imgb4');
  const img5 = document.getElementById('imgb5');
  // Parallaxe au scroll
  $(window).on("scroll", function () {
    let scrollPos = $(this).scrollTop();

    // Déplacement des éléments
    $("#imgb5").css("left", 500 + scrollPos * 1.3 + "px");
    $("#imgb4").css("left", 500 + scrollPos * 1 + "px");
    $("#imgb3").css("left", 500 + scrollPos * 1.75 + "px");
    $("#imgb1").css("left", 500 + scrollPos * 1.5 + "px");

    // Cache les éléments après un certain scroll
    if (scrollPos > 500) {
      $("#imgb1, #imgb3, #imgb4, #imgb5").fadeOut();
    } else {
      $("#imgb1, #imgb3, #imgb4, #imgb5").fadeIn();
    }
  });
});
```

J'ai laissé la **position: absolute** mais j'ai rajouté et modifié la **position top** en changeant les pixels en fonction de la **taille des images**.

- #imgb5 avec 540px
- #imgb4 avec 560px
- #imgb1 avec 540px
- #imgb3 avec 530px

```
#imgb2{
  width: 70%;
  padding-bottom: 64px;
}
#imgb5{
  position: absolute;
  top: 540px;
}
#imgb4{
  position: absolute;
  top: 560px;
}
#imgb1{
  position: absolute;
  top: 540px;
}
#imgb3{
  position: absolute;
  top: 530px;
}
```

1/ Avec **\$(window).on("scroll", function() {})** on peut définir une fonction qui sera exécutée à chaque fois que l'utilisateur fait défiler la page. J'ai déclaré une variable **let scrollPos = \$(this).scrollTop();** qui récupère la position actuelle du défilement de la page qu'on peut savoir grâce a **.scrollTop()** renvoie la position du défilement en pixels par rapport au haut de la page.

Le code déplace une image vers la gauche en fonction du défilement de la page. J'ai défini une position initale en **500px + (scrollPos définit au-dessus \* facteur)**, qui représente la vitesse de l'élément qui se déplace. Ainsi, plus l'utilisateur fait défiler la page, plus l'élément se déplace, avec des vitesses différentes selon le facteur.

2/ J'ai rajouté une partie qui fait disparaître les img après un certain seuil de défilement. La condition **if (scrollPos > 500)** vérifie si la position du défilement dépasse **500px**.

Si scrollPos > 500, les éléments **#imgb1, #imgb3, #imgb4, et #imgb5** disparaîtront grâce à **.fadeOut()**. Si la position de défilement est inférieure à 500 pixels, les éléments redeviennent visibles avec **.fadeIn()**.

# Parallaxe

Le code permet de créer un effet de parallaxe où les images se déplacent horizontalement avec différentes vitesses à mesure que l'utilisateur fait défiler la page.

En plus de cela, les images disparaissent lorsque la page est suffisamment défilée , et réapparaissent lorsque l'utilisateur fait défiler la page vers le haut.





# Animation 6:

## Apparition d'image:

```
//Animer la poubelle le gaz et la terre
if ($(this).scrollTop() > 2200) {
    $(".imgs1").fadeIn(500);
}

if ($(this).scrollTop() > 3200) {
    $(".imgs2").fadeIn(500);
}

if ($(this).scrollTop() > 4000) {
    $(".imgs3").fadeIn(500);
}
```



Pour **.imgs1**, j'ai mis left: **78%** et ajusté la largeur à **20%**.  
Pour **.imgs2**, j'ai placé l'image à **top: 3500px** et **left: 1.5%** avec une largeur de **18%**.  
Pour **.imgs3**, la position est à **top: 4200px** et left: **80%** avec une largeur de **15%**.

J'ai laissé la position absolute, mais j'ai modifié la position left et top pour chaque image en ajustant les valeurs selon leur emplacement.

Le début du code vérifie à chaque défilement de la page, le code examine la distance parcourue par rapport au haut de la page en utilisant **\$(this).scrollTop()**.

- 1/ Si on fait défiler défiler la page de plus de **2200 pixels**, il fait apparaître l'élément avec la classe **.imgs1** en utilisant un effet de fondu (fadeIn) pendant **500 ms**.
- 2/ Si on fait défiler la page de plus de **3200 pixels**, il fait apparaître l'élément avec la class **.imgs2** en utilisant également un effet de fondu de **500 ms**.
- 3/ Si on fait défiler défiler la page de plus de **4000 pixels**, il fait apparaître l'élément avec la classe **.imgs3** en utilisant un effet de fondu de **500 ms**.

```
.imgs1{
    position: absolute;
    left: 78%;
    width: 20%;
    z-index: 1;
}

.imgs2{
    position: absolute;
    top: 3500px;
    left: 1.5%;
    width: 18%;
    z-index: 1;
}

.imgs3{
    position: absolute;
    top: 4200px;
    left: 80%;
    width: 15%;
    z-index: 1;
}

.imgs1, .imgs2, .imgs3 {
    display: none;
}
```

# Apparition d'image

Lorsque l'utilisateur fait défiler la page, chaque condition if vérifie la position de défilement.

Si l'utilisateur a fait défiler la page jusqu'à un certain point (2200px, 3200px, ou 4000px), l'élément correspondant avec la classe **.imgs1**, **.imgs2**, ou **.imgs3** devient visible avec une animation de fondu.

Cela permet d'afficher ces éléments au fur et à mesure que l'utilisateur descend sur la page, créant un effet dynamique où les éléments "apparaissent" pendant le défilement.



Ci-dessous le lien qui montrant l'ingratitude de mes animations en **Jquery**

**LIEN**

J'ai aussi déposé sur e-campus Un dossier en format ZIP contenant le HTML, CSS et le JS

**PS: Charger la page en 100%**

**RALAIKOA** Samuel  
MMI Groupe B1