

ANIMATIONS EN CSS

Insérer Gif

OBJECTIFS

- ➔ Comprendre ce qu'est une animation CSS
- ➔ Connaître les règles de la propriété *@keyframes* permettant de gérer l'animation
- ➔ Savoir utiliser ces règles pour les introduire dans un site

SITES DE RÉFÉRENCE

➔ https://www.w3schools.com/css/css3_animations.asp

➔ <https://developer.mozilla.org/fr/docs/Web/CSS/@keyframes>

➔ https://developer.mozilla.org/fr/docs/Web/CSS/Animations_CSS/Utiliser_les_animations_CSS

➔ https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions

➔ <https://developer.mozilla.org/fr/docs/Web/CSS/transition>

1.

Qu'est-ce que qu'une
animation css

Définition

- ➡ La CSS permet l'animation d'éléments HTML sans utiliser JavaScript ou Flash!
- ➡ Une animation permet à un élément changer progressivement d'un style à l'autre.
- ➡ Vous pouvez changer autant de propriétés CSS que vous voulez, autant de fois que vous voulez.
- ➡ Pour utiliser l'animation CSS, il est nécessaire d'utiliser la règle « *@keyframes* » .
- ➡ La règle *@keyframes* permet aux auteurs de définir les étapes qui composent la séquence d'une animation CSS.
- ➡ Cela permet de contrôler une animation plus finement que ce qu'on pourrait obtenir avec les *transitions*.

Les propriétés de @keyframes

- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation-fill-mode
- animation

2.

Les règles de base de
« @keyframes »

Présentation

- ➔ Lorsque vous spécifiez des styles CSS dans les règles de **@keyframes**, l'animation passera progressivement du style actuel au nouveau style.
- ➔ Pour faire fonctionner une animation, il faut lier une animation à un élément.
- ➔ Vous pouvez changer autant de propriétés CSS que vous voulez, autant de fois que vous voulez.
- ➔ Pour utiliser l'animation CSS, il est nécessaire d'utiliser des « **keyframes** » spécifiques (mots clés).
- ➔ Les « **keyframes** » permettent de définir quel style aura un élément à un moment donné.

Exemple 1

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: 2s;
      }
      @keyframes example {
        from {background-color: red;}
        to {background-color: yellow;}
      }
    </style>
  </head>
  <body>
    <p><b>Note:</b> This example does not work in Internet Explorer 9 and
earlier versions.</p>
    <div class="carre"></div>
  </body>
</html>
```

Remarques

- ➔ La propriété ***animation-duration*** permet de définir le temps total pendant lequel se déroule l'animation.
- ➔ Si la propriété ***animation-duration*** n'est pas spécifiée, aucune animation ne se produira car la valeur par défaut est 0s.
- ➔ Dans l'exemple précédent, les attributs « ***from*** » et « ***to*** » définissent respectivement le style de début de l'animation (à **0 %** de celle-ci) et le style de fin de l'animation (à **100 %** de celle-ci).
- ➔ Il est possible d'utiliser la propriété **%** pour définir un nombre illimité de transition de style...

Exemple 2 : Plusieurs transitions de styles

On définit
ici 4
transitions
en 4s

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
      }
      @keyframes example {
        0% {background-color: red;}
        25% {background-color: yellow;}
        50% {background-color: blue;}
        100% {background-color: green;}
      }
    </style>
  </head>
  <body>
    <p><b>Note:</b> This example does not work in
    Internet Explorer 9 and earlier versions.</p>
    <div class="carre"></div>
    <p><b>Note:</b> When an animation is finished,
    it changes back to its original style.</p>

  </body>
</html>
```

Remarques

- ➔ Il est aussi possible de modifier à la fois la couleur du fond de la **div** « carre », mais aussi sa position !!
- ↳ Préciser dans la css de la **div** « carre » que la propriété **position** de cette div est **relative**.
- ↳ Dans **@keyframes** : utiliser la propriété « **left** » pour définir la **position** suivant l'axe **x** de la div
- ↳ Dans **@keyframes** : utiliser la propriété « **top** » pour définir la **position** suivant l'axe **y** de la div

Exemple 3 : Transition de style et mouvement

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>
    <p><b>Note:</b> This example does not work in Internet
    Explorer 9 and earlier versions.</p>
    <div class="carre"></div>
    <p><b>Note:</b> When an animation is finished, it
    changes back to its original style.</p>
  </body>
</html>
```

3.

**Retarder une
animation**

Présentation

- Il est possible retarder une animation en utilisant la propriété « **animation-delay** »
- ↳ Spécifier le temps de délai en secondes...

Exemple 4 : retarder le début de l'animation

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: 2s;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>

    <div class="carre"></div>

  </body>
</html>
```


Présentation

- Des valeurs négative à la propriété « **animation-delay** » sont aussi acceptées
- ↳ Lorsque l'animation commence tout se passe comme si elle s'était lancée il y a N secondes.

Exemple 5 : avancer le début de l'animation (retard négatif!!)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>

    <div class="carre"></div>

  </body>
</html>
```

4.

**Réglage du nombre
d'itération d'une
animation**

Présentation

→ La propriété

« animation-iteration-count »

permet de spécifier le nombre d'itération de l'animation.

Exemple 6 : réglage du nombre d'itération

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
        animation-iteration-count: 3;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>

    <div class="carre"></div>

  </body>
</html>
```

Présentation

- Si la propriété
« animation-iteration-count »
prend la valeur « infinite »,
l'animation s'exécute en continu
tant que la page n'est pas
fermée.

Exemple 7 : Faire tourner l'animation de façon continue à l'infini.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
        animation-iteration-count: infinite;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>

    <div class="carre"></div>

  </body>
</html>
```

5.

Faire tourner une

animation en

sens inverse

ou

cycles alternatifs

Présentation

→ La propriété

« animation-direction » permet de spécifier le sens d'exécution d'une animation :

- Sens direct
- Sens inverse
- De façon alternative (aller-retour)

→ La propriété « **animation-direction** » peut prendre les valeurs suivantes :

- **normal** : valeur par défaut. L'animation tourne dans la sens direct.
- **reverse** : l'animation tourne à l'envers.
- **alternate** : l'animation tourne d'abord dans le sens direct et ensuite en sens inverse.
- **alternate-reverse** : l'animation tourne d'abord dans le sens inverse et ensuite en sens direct.

Exemple 8 : Faire tourner l'animation en sens inverse.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
        animation-iteration-count: infinite;
        animation-direction: reverse;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

Exemple 9 : Faire tourner l'animation d'abord en sens direct et ensuite en sens inverse.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
        animation-iteration-count: infinite;
        animation-direction: alternate;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

Exemple 10 : Faire tourner l'animation d'abord en sens inverse et ensuite en sens direct.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 4s;
        animation-delay: -2s;
        animation-iteration-count: infinite;
        animation-direction: alternate-reverse;
      }
      @keyframes example {
        0% {background-color:red; left:0px; top:0px;}
        25% {background-color:yellow; left:200px; top:0px;}
        50% {background-color:blue; left:200px; top:200px;}
        75% {background-color:green; left:0px; top:200px;}
        100% {background-color:red; left:0px; top:0px;}
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

6.

**Spécifier la dynamique
(vitesse et accélération)
du mouvement de
l'animation**

→ La propriété

« animation-timing-function »

permet de spécifier la

dynamique d'une animation

➔ La propriété « **animation-timing-function** » peut prendre les valeurs suivantes :

ease : départ lent, accélération et fin lente.

linear : vitesse constante.

ease-in : départ lent ensuite vitesse constante.

ease-out : vitesse constante et fin lente.

ease-in-out : départ et fin lents et vitesse constante au milieu.

cubic-bezier(n,n,n,n) : permet de définir les différents temps d'accélération et décélération grâce à une fonction cubique de Bezier.

Exemple 11 : Choisir la dynamique de mouvement de l'animation.

Équivalent à :
animation-name : mymove ;
animation-duration: 5s ;
animation-iteration-count: infinite;



```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        width: 100px;
        height: 50px;
        background-color: red;
        font-weight: bold;
        position: relative;
        animation: mymove 5s infinite;
      }
      /* Standard syntax */
      #div1 { animation-timing-function: linear; }
      #div2 { animation-timing-function: ease; }
      #div3 { animation-timing-function: ease-in; }
      #div4 { animation-timing-function: ease-out; }
      #div5 { animation-timing-function: ease-in-out; }
      /* Standard syntax */
      @keyframes mymove {
        from {left: 0px;}
        to {left: 300px;}
      }
    </style>
  </head>
  <body>
    <p><strong>Note:</strong> The animation-timing-funtion
    property is not supported in Internet Explorer 9 and earlier versions.</p>
    <div id="div1">linear</div>
    <div id="div2">ease</div>
    <div id="div3">ease-in</div>
    <div id="div4">ease-out</div>
    <div id="div5">ease-in-out</div>
  </body>
</html>
```

7.

Spécifier le mode de
remplissage d'une
animation

→ Une animation n'affecte pas un élément avant ou après qu'une propriété soit lancée :

↳ la propriété
« animation-fill-mode »
permet de passer outre ce
comportement

➔ La propriété « **animation-fill-mode** » spécifie un style pour l'élément cible lorsque l'animation n'est pas lancée :

- avant qu'elle commence,
- après qu'elle ait terminé,
- ou les deux

➔ La propriété « **animation-fill-mode** » prend les valeurs suivantes :

none : valeur par défaut. L'animation n'applique aucun style avant ou après qu'elle soit exécutée.

forwards : l'élément conserve les valeurs de style qui sont définis par la dernière propriété de *@keyframes* (dépend de *animation-direction* and *animation-iteration-count*).

backwards : l'élément prend les valeurs de style qui sont définis par la première propriété de *@keyframes* (dépend de *animation-direction*) et conserve cette propriété toute la durée de l'animation (*animation-delay*).

both : l'animation suit les règles des propriétés **forwards** et **backwards** et étend les propriétés dans les deux directions.

Exemple 12 : Conserver le style de la dernière keyframe à la fin de l'animation.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 3s;
        animation-fill-mode: forwards;
      }
      @keyframes example {
        from {top:0px;}
        to {top:200px ; background-color:blue}
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

Exemple 13 :
le carré prend le style de la
première keyframe tant
que l'animation n'a pas
commencé pendant un
temps défini par la
propriété *animation-delay*.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 3s;
        animation-delay: 2s;
        animation-fill-mode: backwards;
      }
      @keyframes example {
        from {top:0px;background-color:yellow}
        to {top:200px ; }
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

Exemple 14 :
le carré prend le style de la
première keyframe tant
que l'animation n'a pas
commencé pendant un
temps défini par la
propriété *animation-delay*.
Et garde le style de la
dernière keyframe lorsque
l'animation se termine.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .carre{
        width: 100px;
        height: 100px;
        Position: relative ;
        background-color: red;
        animation-name: example;
        animation-duration: 3s;
        animation-delay: 2s;
        animation-fill-mode: both;
      }
      @keyframes example {
        from {top:0px;background-color:yellow}
        to {top:200px ; }
      }
    </style>
  </head>
  <body>
    <div class="carre"></div>
  </body>
</html>
```

8.

Écriture condensée de
la propriété animation

→ Les écritures suivantes sont équivalentes :

```
<style>
  .carre{
    animation-name: example;
    animation-duration: 5s;
    animation-timing-function: linear;
    animation-delay: 2s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
  }
</style>
```

```
<style>
  .carre{
    animation : example 5s linear 2s infinite alternate;
  }
</style>
```