

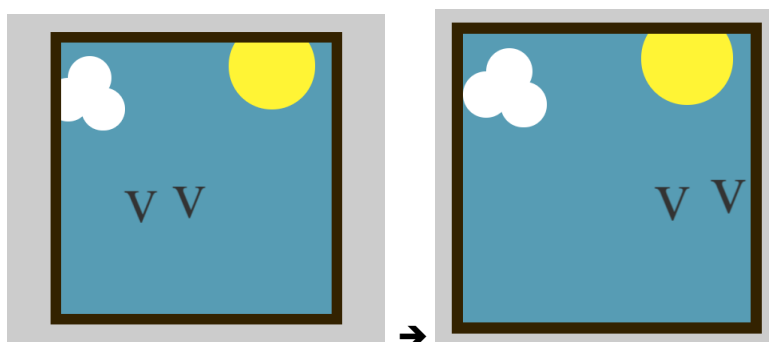
# Vol au-dessus d'un nid de coucou

TP/tutoriel animation CSS

Dans ce TP-tuto, nous allons réaliser un décor tout à fait charmant et pittoresque en HTML et CSS.

- 0 image
- 0 javascript
- 0 gluten

Voilà ce que l'on aura à la fin :



Contenez votre émotion. Il s'agit comme vous l'avez compris d'une « fenêtre » au travers de laquelle on voit passer au gré du temps un nuage et deux oiseaux... Cela ne se voit pas ici, mais notre œuvre d'art est animée et [vous pouvez le constater ici](#). Alors c'est parti !

## Le HTML de notre page

Nous allons réaliser une page HTML très simple. Elle doit simplement décrire la fenêtre, un nuage, deux oiseaux et le soleil. Le résultat pourra ressembler à ça :

```
<html>
<head></head>
<body>
<div class="window">
  <span id="sun">o</span>
  <span class="cloud">o</span>
  <span class="bird">v</span>
  <span class="bird">v</span>
</div>
</body>
</html>
```

Les « v » vont être utilisés pour faire des oiseaux pas chers.

## Mise en place de notre scène

### La fenêtre et le mur

Nous voulons centrer la fenêtre dans la page, lui donner une taille convenable, appliquer une bordure (pour le cadre) et lui appliquer un joli fond bleu. Avec ce que l'on a vu jusque-là, cela devrait être simple.

```
body {  
    background: #ccc;  
}  
/* Notre jolie "fenetre" */  
.window {  
    border : 8px solid #320;  
    position: relative;  
    height: 200px;  
    width: 200px;  
    overflow: hidden;  
    margin : 30px auto;  
    background: #579CB4;  
}
```

### Le soleil

Rendons notre soleil un peu plus probant. Pour cela, on doit lui appliquer une couleur de fond, l'arrondir masquer le texte et le positionner. Là aussi vous devriez savoir le faire.

```
#sun {  
    color: transparent;  
    position: absolute;  
    display: inline-block;  
    background : #FFF435;  
    width: 64px;  
    height: 64px;  
    border-radius : 32px;  
    top: -15px;  
    right: 12px;  
}
```

### Maintenant notre nuage...

Cela se complique un peu mais vous êtes aussi censé pouvoir y arriver. On va se baser sur le même principe que pour le soleil (faire un cercle coloré sans texte lisible) dans un premier temps :

```
/* Création et animation de notre nuage */  
.cloud {  
    color: transparent;  
    position: relative;  
    display: inline-block;  
    background : #fff;  
    width: 32px;  
    height: 32px;  
    border-radius : 16px;  
}
```

Cela ne nous donne alors qu'un cercle blanc, alors que l'on en veut trois.

On va utiliser les pseudo-classes :before et :after pour reproduire l'opération :

```
.cloud:before {
  content: '<span>&nbsp;</span>';
  color: transparent;
  position: relative;
  display: inline-block;
  background : #fff;
  width: 32px;
  height: 32px;
  border-radius : 16px;
  top: 16px;
  left: -16px;
}
.cloud:after {
  content: '<span>&nbsp;</span>';
  color: transparent;
  position: relative;
  display: inline-block;
  background : #fff;
  width: 32px;
  height: 32px;
  border-radius : 16px;
  top: -32px;
  left: 10px;
}
```

**Votre travail :** on peut améliorer ça. Ici, on répète énormément de propriétés alors que l'on peut faire bien mieux. A vous de faire le nécessaire !

## Et enfin, les oiseaux

Ici, le travail est relativement simple, on ne fait rien de spécial. (C'est peut-être pour cela qu'ils sont si moches)

```
.bird {
  display: inline-block;
  position : relative;
  color: #333;
  font-size: 50px;
  left : -50px;
  top: 40px;
}
```

## Move it

On souhaite faire en sorte que notre nuage traverse la fenêtre à intervalle de temps régulier. On veut que les oiseaux en fassent de même, plus lentement, et en oscillant un peu de haut en bas. Ce sera un bon début.

Pour faire cela, et c'est l'objet de ce TP en fait, nous avons à disposition les animations CSS. Il existe aussi les transitions, qui sont plus simples à manipuler mais qui ne nous aideront pas ici.

### Les animations

Vous pouvez jeter un œil sur la très bonne documentation de Mozilla [au sujet des animations](#). Les animations nous permettent de paramétrer :

- Le nom de l'animation  
(nom que nous utiliserons plus tard pour spécifier le comportement de l'animation)
- La durée de l'animation
- Le nombre de répétition de l'animation
- D'autres trucs palpitants

Par exemple, si on veut faire bouger toutes nos div :

```
div {  
  animation-duration: 20s;  
  animation-name: cmondosomessing;  
  animation-iteration-count: infinite;  
}
```

Une fois ces caractéristiques décrites dans notre CSS, rien ne marche. Normal. Il faut créer cette animation (celle dont on a spécifié le nom). Cela se fait aussi en CSS. Cela se fait à l'aide du mot clé **@keyframes** de la façon suivante :

```
@keyframes cmondosomessing { /* animation nommée cmondosomessing */  
  0% {  
    top : 10px;  
    left : -50px;  
  }  
  12% {  
    top : 1px;  
  }  
  100% {  
    top : 10px;  
    left : 250px;  
  }  
}
```

On peut spécifier autant de « keyframes » dans notre animation que l'on veut, c'est-à-dire autant de pourcentage d'avancement de l'animation pour lesquelles on veut spécifier des propriétés CSS. Le navigateur fera ainsi son possible pour animer chacune de vos propriétés en les faisant transiter d'une valeur à une autre (en changeant progressivement la position, la taille, la couleur, l'opacité, etc...). Toutes les propriétés CSS ne sont pas « animables », jetez une nouvelle fois un œil sur le très bon site de Mozilla (cours non sponsorisé) pour la [liste des balises qui vont marcher](#).

Notez que la portion de code ci-dessus se place (description d'une animation via @keyframes), comme tout le reste dans la feuille CSS, indifféremment avant ou après leur utilisation. On peut décrire autant d'animation qu'on le désire, du moment qu'elles n'ont pas le même nom

Revenons à notre scène

## Animation du nuage

Et on commence par faire traverser notre nuage de gauche à droite. Pour cela, on va d'abord affecter les propriétés d'animation pour notre balise nuage :

```
.cloud {  
  /* sum stuff */  
  
  animation-duration: 20s;  
  animation-name: cloud;  
  animation-timing-function: linear;  
  animation-iteration-count: infinite;  
}
```

On utilise ici la propriété CSS *animation-timing-function* qui permet de dire comment l'animation est censé accélérer/ralentir au cours de sa vie (par exemple, de plus en plus vite, ou vite => doucement => vite, etc.). Ici, notre nuage doit se promener pépère, donc sans variation de vitesse : d'où l'utilisation de la valeur « linear ». On répète indéfiniment l'animation qui fera ainsi passer pour la vie un nuage de gauche à droite.

Notez qu'il n'est pas utile de spécifier l'animation pour les pseudo-classes before et after, puisque ce que l'on change va affecter les propriétés de .cloud et donc de ce que l'on décrit dans le before et le after.

Si vous avez bien suivi, il faut que l'on décrive maintenant l'animation correspondante (passage de gauche à droite).

```
@keyframes surprise {  
  0% {  
    top : 10px;  
    left : -50px;  
  }  
  100% {  
    top : 10px;  
    left : 250px;  
  }  
}
```

Si vous lancez ça, ça ne marche pas. Il y a un truc à changer, et c'est très simple, donc activez le neurone.

## Animation des oiseaux

Dernière étape. On va faire bouger nos oiseaux, également de la gauche vers la droite, mais en plus, il va falloir les faire osciller.

Commençons une nouvelle fois par leur affecter une animation :

```
.bird {  
  /* sum stuff */  
  animation-duration: 6s;  
  animation-name: fly;  
  animation-timing-function: linear;  
  animation-iteration-count: infinite;  
}
```

Et il nous reste alors à décrire l'animation sus-utilisée. Ici, c'est beaucoup plus long mais pas plus compliqué que précédemment :

```
@keyframes lol {  
  0% {  
    transform: translate(0);  
  }  
  5% {  
    transform : translate(20px, 5px);  
  }  
  15% {  
    transform : translate(40px, -5px);  
  }  
  25% {  
    transform : translate(60px, 5px);  
  }  
  35% {  
    transform : translate(80px, -5px);  
  }  
  45% {  
    transform : translate(100px, 5px);  
  }  
  55% {  
    transform : translate(120px, -5px);  
  }  
  65% {  
    transform : translate(140px, 5px);  
  }  
  75% {  
    transform : translate(160px, -5px);  
  }  
  85% {  
    transform : translate(180px, 5px);  
  }  
  95% {  
    transform : translate(200px, -5px);  
  }  
  100% {  
    transform : translate(220px);  
  }  
}
```

Et pourquoi ça marche pas ? Et oui, même raison que pour l'animation précédente, donc on corrige.

Et on se rend compte que nos oiseaux volent un peu trop en formation... Ce serait bien de rendre leur trajectoire un brin moins synchro. Pour cela, nous allons donc utiliser la même animation, mais nous allons la lancer avec un petit décalage pour l'un de nos deux oiseaux, afin de donner un air d'indépendance entre les deux. Pour faire cela, nous allons utiliser la propriété CSS dédiée `animation-delay` :

```
.bird:???? {  
  animation-delay: 0.3s;  
}
```

A vous de vous rappeler comment on sélectionne un n-ième élément pour un sélecteur donné afin de profiter de cette formidable évolution dans notre animation.

## Améliorations

Il serait super classe de mettre en place un cycle jour/nuit. Pour cela, soyez créatif mais une piste de travail pour les gens pas amusants :

- Faites descendre le soleil en dessous du niveau de la fenêtre puis faites le remonter (en ligne droite, on vous demande pas de faire des calculs d'écliptique)  
Pour faire cela, vous pouvez créer une animation qui fait simplement descendre le soleil et dire au navigateur de la jouer à l'endroit, puis à l'envers, le tout en conservant une itération infinie. La propriété permettant de spécifier que l'on veut jouer l'animation une fois dans un sens, une fois dans l'autre, est la suivante :  
`animation-direction: alternate;`
- Une fois que votre soleil se couche et se lève, il faudrait changer la couleur du ciel sur le même rythme. Pour cela il vous suffit de spécifier une animation pour la fenêtre, en changeant simplement sa couleur de fond sur un rythme alterné (comme le soleil).
- Si vous êtes bon, vous pourrez faire lever la lune en même temps que le soleil se couche et nous placer deux trois étoiles, mais n'en demandons pas trop !
- Ce qui serait vraiment classe, ce serait que cette animation (composée en fait de trois animations : soleil, lune, fenêtre) ne soit pas linéaire comme celle des nuages et oiseaux. Allez, vous arriverez à faire au moins quelque [chose comme ça](#) non ?

Vous pouvez évidemment utiliser des images que vous animerez. Ici, le but est de travailler le CSS, mais utilisez toutes les ressources à votre disposition.