

# Séance n°3

## Animation CSS

### Problématique

On veut créer les **barres de progression** ci-dessous :

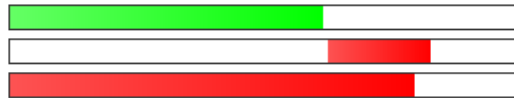


Figure 1, Barres de progression

### Squelette de la page

On donne le code HTML correspondant :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Animation</title>
</head>
<body>
  <div class="loader animation1">
    <div class="progress-bar"></div>
  </div>
  <div class="loader animation2">
    <div class="progress-bar"></div>
  </div>
  <div class="loader animation3">
    <div class="progress-bar"></div>
  </div>
</body>
</html>
```

Le chargement du fichier dans le navigateur n'affiche rien. Il faudra écrire le code CSS.

### Code CSS

Commençons par définir le cadre des barres de progression :

- Dimensions : 320px × 16px
- Marges verticales: 5px
- Bordure : couleur ■ #333, épaisseur 1px

La barre de progression est définie comme suit :

- Dimensions : 100% × 100%
- Arrière plan : Un dégradé linéaire de ■ #007700 à ■ #65ff65, de gauche à droite

```
.loader {
  width: 320px;
  height: 16px;
  margin: 5px 0;
  border: #333 solid 1px;
  position: relative;
}
```

```
.animation1 .progress-bar {
  width: 100%;
  height: 100%;
  background: linear-gradient(to right, #007700, #65ff65);
}
```

Ce code permet d'obtenir un rectangle rempli par un dégradé de vert comme suit :

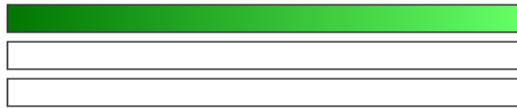


Figure 2, Implémentation partielle de l'indicateur de téléchargement

## Présentation

Une animation permet à un élément de passer progressivement d'un style à un autre. Pour animer un élément, il faut définir les images clés (keyframes) de l'animation. Les images clés contiennent les styles que l'élément aura à certains moments.

### La règle @keyframes

La règle `@keyframes` définit les moments auxquels l'animation passe du style actuel à un nouveau style.

Pour fonctionner les images clés `@keyframes` doivent être liés à un élément de la page.

Dans l'exemple suivant l'animation `anim01` est liée à l'élément ayant comme classe `.anim01`. L'animation dure 4 secondes et fait changer l'arrière plan de l'élément du ■ rouge au ■ jaune.

La propriété `animation-duration` définit la durée d'exécution de l'animation. Si elle n'est pas spécifiée, aucune animation ne se produit, car la valeur par défaut est 0s.

Dans cet exemple, les moments quand le style changera en utilisant les mots clés `from` et `to`.

- `from` : représente le style au début de l'animation (0%)
- `to` : représente le style à la fin de l'animation (100%)

```
@keyframes anim01 {
  from {background-color: red;}
  to {background-color: yellow;}
}

.anim01 {
  width: 100px; height: 100px;
  background-color: red;
  animation-name: anim01;
  animation-duration: 4s;
}
```

À la place des mots clés `from` et `to`, il est possible d'utiliser le pourcentage de progression de l'animation « **animation progression in percent** » :

- `0%` représente le début de l'animation,
- `x%` avec `x` est un nombre de 0 à 100 qui représente la progression de l'animation,
- `100%` représente la fin de l'animation

**Exemple 1** : L'exemple suivant modifie la couleur d'arrière-plan de la classe `.anim02` :

- de `0%` à `25%` : couleur d'arrière-plan ■ rouge
- de `25%` à `50%` : couleur d'arrière-plan ■ jaune
- de `50%` à `75%` : couleur d'arrière-plan ■ bleue
- de `75%` à `100%` : couleur d'arrière-plan ■ verte

```
@keyframes anim02 {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

.anim02 {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: anim02;
  animation-duration: 4s;
}
```

**Exemple 2 :** L'exemple suivant modifie la couleur et la position de l'élément dont la classe est `.anim03`.

```
@keyframes anim03 {
  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;}
}

.anim03 {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: anim03;
  animation-duration: 4s;
}
```

## Propriétés d'une animation

Une animation possède les propriétés suivantes :

- `animation-name` nom donné aux images clés (`@keyframes`)
- `animation-duration` durée de l'animation en secondes
- `animation-delay` permet de lancer l'animation avec un retard donné
- `animation-iteration-count` nombre de répétition de l'animation
- `animation-direction` indique le sens d'évolution de l'animation de 0% vers 100% ou le contraire.
- `animation-timing-function` indique la vitesse d'évolution de l'animation
- `animation` est un raccourci à l'ensemble des propriétés précédentes dans une seule propriété :

```
animation: name duration timing-function delay iteration-count direction;
```

**Exemple 1 :**

```
.anim04 {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: anim04;
  animation-duration: 10s;
  animation-delay: 5s;
  animation-iteration-count: 6;
  animation-direction: alternate;
}
```

```
@keyframes anim04 {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; border-radius: 50%; left:200px; top:0px;}
  50% {background-color:blue; border-radius: 0%; left:200px; top:200px;}
  75% {background-color:green; border-radius: 50%; left:0px; top:200px;}
  100% {background-color:red; border-radius: 0%; left:0px; top:0px;}
}
```

### Exemple 2 :

```
.anim05 {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation: anim05 4s linear 5s 7 alternate;
}
```

## Application 4 : Progress bar

### Travail demandé

1. Ajouter une animation qui fait varier la largeur du `.progress-bar` de la première animation `.animation1` de 0% à 100% :
  - Nommer l'animation `animation1`.
  - Fixer la durée de l'animation à 5s.
  - Utiliser `linear` comme `timing-function`.
  - Répéter l'animation une infinité de fois `infinite`.

#### Code CSS

```
.animation1 .progress-bar {
  /* ajouter les lignes suivantes */
  animation-name: animation1;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
}

@keyframes animation1 {
  from { width: 0%; }
  to { width: 100%; }
}
```

2. Copier le code CSS de `.animation2 .progress-bar` :

```
.animation2 .progress-bar {
  width: 64px;
  height: 100%;
  background: linear-gradient(to left, #f00, #fd5353);
  position: absolute;
}
```

Quel est le résultat de ce code ?

3. Déplacer le rectangle rouge de gauche à droite, puis de gauche à droite une infinité de fois. Pour faire, utiliser la propriété CSS `left`.

#### Code CSS

```
.animation2 .progress-bar {
  width: 64px;
  height: 100%;
  background: linear-gradient(to left, #f00, #fd5353);
  position: absolute;

  animation-name: animation2;
  animation-duration: 5s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
}

@keyframes animation2 {
  from { left: 0; }
  to { left: 256px; }
}
```

4. Copier le code CSS des images clés de l'animation `animation3` :

```
@keyframes animation3 {
  0% {
    background: linear-gradient(to left, #00f, #000070);
    width: 0;
  }

  50% {
    background: linear-gradient(to left, #ff9900, #ffba53);
    width: 50%;
  }

  100% {
    background: linear-gradient(to left, #f00, #fd5353);
    width: 100%;
  }
}
```

5. Ajouter le code CSS du sélecteur `.animation3 .progress-bar` :
- Dimensions : 100% × 100%, positionnement : `absolute`
  - Nom de l'animation : `animation3`, durée : `2.5s`, répétition : `infinite`
  - Direction : `alternate`
  - Fonction de timing : `ease-in-out`

```
.animation3 .progress-bar {
  width: 100%;
  height: 100%;
  position: absolute;

  animation-name: animation3;
  animation-duration: 2.5s;
  animation-direction: alternate;
  animation-timing-function: ease-in-out;
  animation-iteration-count: infinite;
}
```

## Application 5 : Smiley

On désire créer le smiley suivant :



Figure 3, Forme initiale du smiley

### Travail demandé

1. Copier le code HTML suivant et faire la mise en forme du smiley à l'aide du CSS.

```
<div id="tete">
  <div id="yeux">
    <div class="oeil">
      <div class="iris">
        <div class="pupille"></div>
      </div>
    </div>
    <div class="oeil">
      <div class="iris">
        <div class="pupille"></div>
      </div>
    </div>
  </div>
  <div class="bouche"></div>
</div>
```

2. Animer les yeux du smiley et sa bouche.

## Application 6 : Montre Analogique

On désire créer une montre analogique comportant deux aiguilles, comme suit.

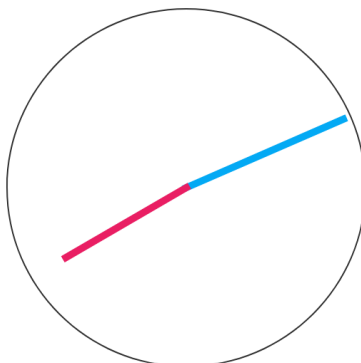


Figure 4, Montre analogique à deux aiguilles

## Travail demandé

1. Copier le code HTML. Faire la mise en forme des deux aiguilles.

```
<div class="horloge">
  <div class="aiguille1"></div>
  <div class="aiguille2"></div>
</div>
```

2. Placer les aiguilles au centre de la montre.
3. Animer les aiguilles de la montre : la petite aiguille doit être 12 fois moins lente que la plus longue. La montre tourne indéfiniment sans s'arrêter.

## Exercice de rappel

Créer une animation comme suit :

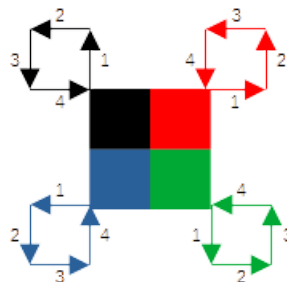


Figure 5, Animation carrés

## Code HTML

On donne le code HTML suivant :

```
<div class="gcarre">
  <div class="carre carre1"></div>
  <div class="carre carre2"></div>
  <div class="carre carre3"></div>
  <div class="carre carre4"></div>
</div>
```

## Code CSS

On code CSS suivant, qui permet de dessiner les carrés :

```
.gcarre {
  width: 200px; height: 200px;
  position: relative;
  margin: calc(50vh - 100px) auto;
}
.gcarre > .carre {
  width: 100px; height: 100px;
  position: absolute;
}
.carre1 { background-color: black; top: 0; left: 0; }
.carre2 { background-color: red; top: 0; left: 100px; }
.carre3 { background-color: blue; top: 100px; left: 0; }
.carre4 { background-color: green; top: 100px; left: 100px; }
```

## Travail demandé

Utiliser les propriétés CSS vue dans ce cours pour animer ces carrés selon le cycle indiqué dans l'image : `@keyframes`, `animation-name`, `animation-duration`, `animation-iteration-count` et `animation-timing-function`.