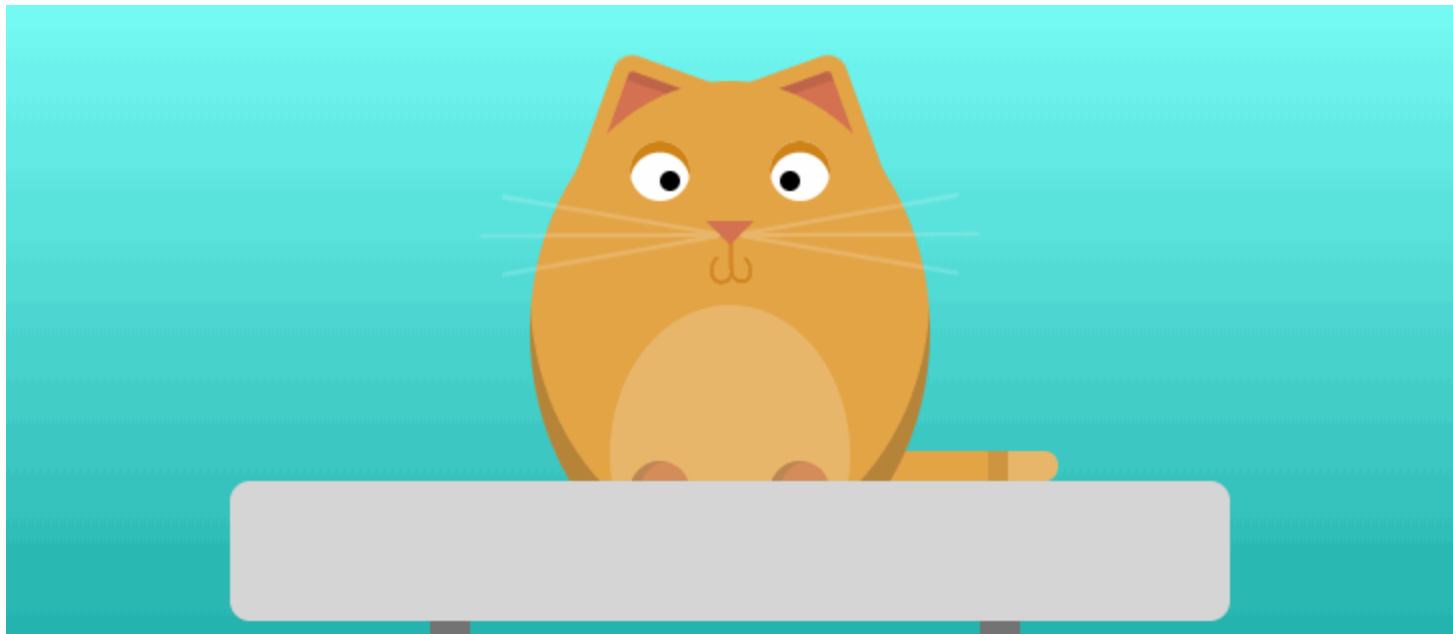


# Les Animations en CSS



# L'arrivée de CSS3

Le CSS permet de définir l'ensemble de la **mise en page des pages web**, comme :

- les couleurs,
- les polices,
- les tailles de caractères
- et tous les éléments qui participent au design d'une page.

Le CSS offre **une grande souplesse d'utilisation**, permettant de changer facilement la structure et la présentation d'un site web.

L'avènement du CSS3 a révolutionné la création des sites en simplifiant l'écriture des pages web.

# Animation vs Transition

Transitions et animations se différencient sur le fait que les animations laissent plus de libertés et plus contrôle sur le déclenchement et la progression du changement de valeur des propriétés que l'on souhaite animer.

# Utilisation de « Transitions » CSS

Les transitions CSS permettent de modifier les propriétés CSS en douceur et pendant un certain temps.

Ainsi, vous pouvez contrôler le processus de transition des éléments d'un état à un autre.

<file:///C:/Users/Acs/Dogger/index.html>

# L'avantage des Animations

Les animations vous permettent de faire des animations plus complexes que des transitions.

Les animations permettent de modifier la valeur d'une propriété progressivement mais en utilisant « les keyframes ».

Qui permettent de gérer le comportement des animations en définissant :

- la durée d'animation,
- le nombre de répétition
- le comportement de répétition.

# Utilisation des « Animations » CSS

Une animation CSS est toujours définie par une règle `@keyframes` suivie du nom de votre animation. C'est ensuite à l'intérieur de ce bloc que vous indiquerez les étapes de l'animation et les propriétés des différents éléments.

```
@keyframes mon-animation {  
    From {  
        To {  
        }  
    }  
}
```

Dans les blocs « `From` » et « `To` » vous devez renseigner les règles à appliquer

# KEY FRAME

| Valeur             | Description  |
|--------------------|--|
| animationname      | Spécifie le nom de l'animation   |
| keyframes-selector | Spécifie le pourcentage de la durée de l'animation :<br>-from (même que 0%)<br>-to (même que 100%) |
| css-styles         | Propriétés des styles CSS  |
| initial            | La valeur initiale.  |
| inherit            | La propriété de son élément parent   |

# Les 9 propriétés des Animations CSS

Les propriétés qui vous permettant de contrôler les animations :

- 1- Nom de l'animation « **animation-name** »  
*spécifie le nom de l'animation, qui relie la règle @keyframes à un sélecteur ;*
- 2- Durée d'animation « **animation-duration** » ;
- 3- Fonction de synchronisation d'animation « **animation-timing-function** » ;



4- Retard d'animation « **animation-delay** » ;

5- Nombre d'itérations d'animations « **animation-iteration-count** »  
*spécifie le nombre de répétitions de l'animation,*  
*(la valeur par défaut est 1).*  
*la valeur « infinite » signifie que l'animation sera jouée indéfiniment ;*

6- Animation-direction « **animation-direction** »  
*définit la direction de l'animation ;*

## 7- Etat de lecture d'animation « **animation-play-state** »

*Cette propriété contrôle l'arrêt et la lecture de l'animation. Il y a deux valeurs, « en cours d'exécution » (l'animation est lue, elle est activée par défaut) et « en pause » (l'animation s'arrête) ;*

## 8- Mode de remplissage d'animation ;

## 9- Animation

*La propriété « animation » correspond à la version raccourcie ou notation short-hand des propriétés « animation-\* » vues ci-dessus.*

*Nous allons pouvoir lui passer les différentes valeurs relatives aux propriétés animation- pour créer simplement des animations complètes ;*

# Répéter une animation

Pour que l'animation se répète, il suffit d'utiliser la propriété [animation-iteration-count](#) et d'indiquer le nombre de répétitions souhaitées

Ici, on utilise la valeur « infinite » pour que l'animation se répète à l'infini :

```
p { animation-duration: 3s;  
      animation-name: slidein;  
      animation-iteration-count: infinite; }
```

[https://yari-demos.prod.mdn.mozit.cloud/fr/docs/Web/CSS/CSS\\_Animations/Using\\_CSS\\_animations/sample\\_R%C3%A9p%C3%A9ter\\_une\\_animation.html](https://yari-demos.prod.mdn.mozit.cloud/fr/docs/Web/CSS/CSS_Animations/Using_CSS_animations/sample_R%C3%A9p%C3%A9ter_une_animation.html)

# Obtenir un effet aller-retour

Si on veut que le texte parcourt l'écran de droite à gauche puis de gauche à droite,  
on pourra utiliser la propriété [animation-direction](#) avec la valeur `alternate` :

```
p { animation-duration: 3s;  
    animation-name: slidein;  
    animation-iteration-count: infinite;  
    animation-direction: alternate; }
```

[https://yari-demos.prod.mdn.mozit.cloud/fr/docs/Web/CSS/CSS\\_Animations/Using\\_CSS\\_animations/sample\\_.Obtenir un effet aller-retour.html](https://yari-demos.prod.mdn.mozit.cloud/fr/docs/Web/CSS/CSS_Animations/Using_CSS_animations/sample_.Obtenir_un_effet_aller-retour.html)

# Exemples en couleurs

Créer une animation sur les couleurs

Et sur les couleurs et la forme en même

<https://www.pierre-giraud.com/html-css-apprendre-coder-cours/animation/>

Logo animé « Zoom » repose sur un principe de rotation et utilise le symbole de point géographique pour représenter le thème du voyage.



Transformation du mot Google en microphone, en longueur d'onde, en points rebondissants et finalement, tout simplement en ne conservant que la lettre G.



<https://www.web-eau.net/blog/15-animations-css-vraiment-bluffantes>



C'est la fin !



Merci de votre attention