



Chapitre 1

Les animations

Les animations jouent un rôle important dans la création d'applications et de sites Web attrayants. Les utilisateurs s'attendent à des interfaces utilisateur hautement réactives et interactives. L'animation de votre interface, cependant, n'est pas nécessairement simple.

Qu'est-ce qui devrait être animé, quand et quel objectif devrait avoir cette animation ?

De superbes animations ajoutent une couche de plaisir et d'engagement à vos projets Web pour vos utilisateurs. Vous pouvez animer à peu près tout ce que vous aimez, que ce soit des largeurs, des hauteurs, des positions, des couleurs ou des arrière-plans, mais vous devez être conscient qu'elles peuvent causer une baisse potentielle des performances de votre site et que ces animations modifient la personnalité de votre application. Les animations mal choisies peuvent affecter négativement l'expérience utilisateur ou briser le rythme dans un jeu. Les animations doivent donc être à la fois performantes, appropriées et codées avec le plus de simplicité possible.

OBJECTIFS

- Découvrir à quoi sert une animation.
- Découvrir les 12 principes d'une animation.
- Créer une première animation simple.



Leçon 1-A

Un peu d'histoire

Commençons depuis le début. *Qu'est-ce que l'animation ?*

L'animation, telle que décrite sur Wikipédia, est un ensemble de techniques qui, par une succession d'images, donne à un public l'illusion que des objets en réalité immobiles sont dotés de mouvement naturel.

OBJECTIFS

À la fin de ce premier chapitre, vous serez en mesure :

- d'expliquer c'est quoi et à quoi sert une animation.

Une animation n'est rien de plus que des graphiques en mouvement, c'est la visualisation du changement sur une période de temps, et bien qu'il puisse sembler que faire des animations est quelque chose de plutôt nouveau, il existe des preuves découvertes sur de vieilles murales suggérant que, démontrer le mouvement dans les images est quelque chose que l'homme a essayé d'atteindre depuis environ cinq mille ans.



Aujourd'hui, les animations sont présentes dans presque tous les produits qui peuvent être visualisés sur un écran, mais concentrons-nous spécifiquement sur une branche d'animation conçue pour être performante et être vue sur un navigateur Web. Ce type d'animation est ce que nous appelons l'animation Web.

L'animation Web est un **mouvement sur le Web**, qui est réalisé grâce à différentes technologies et techniques, ou par la combinaison de celles-ci. Les graphiques animés, qui se concentrent surtout sur l'animation de texte à l'écran, les animations Sprite, les animations CSS, les animations JavaScript, les animations SVG et les animations WebGL sont parmi les techniques les plus courantes, mais nous pourrions l'étendre pour inclure les animations d'interface utilisateur, les animations de jeu sur le Web et les visualisations de données animées.

D'autres façons de concevoir l'animation Web consistent à utiliser les GIF, les applets Java, Flash et Shockwave, mais ces technologies sont de moins en moins utilisées au cours des dernières années, voire obsolètes.

De nos jours, l'animation Web est partout, presque chaque annonce Web, site Web, application Web ou programme qui s'exécute sur un navigateur, contient une sorte d'animation Web, certaines sont très subtiles et d'autres plutôt assez complexes, mais ce n'était pas toujours comme ça.

Avant l'arrivée des animations Web, les sites Web n'étaient que des documents simples qui ne pouvaient être liés qu'à d'autres documents simples. Voilà à quoi ressemblait le premier site Web jamais créé, à l'époque, même CSS n'existait pas !

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project. [Mailing lists](#) . [Policy](#) . November's [W3 news](#) . [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) . [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [LineMode](#) [X11 Viola](#) [NeXTStep](#) [Servers](#) [Tools](#) [Mail robot](#) [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web.

[Getting code](#)

Getting the code by [anonymous FTP](#) . etc.

Revenons donc là où tout a commencé. En 1987, CompuServe a introduit le Graphics Interchange Format, mieux connu sous son acronyme GIF. L'une des premières utilisations du GIF sur le Web comme outil d'animation a été de donner aux entreprises la possibilité d'améliorer leurs logos en les faisant pivoter à 360 degrés, et donc de donner l'illusion de la 3D à l'écran.

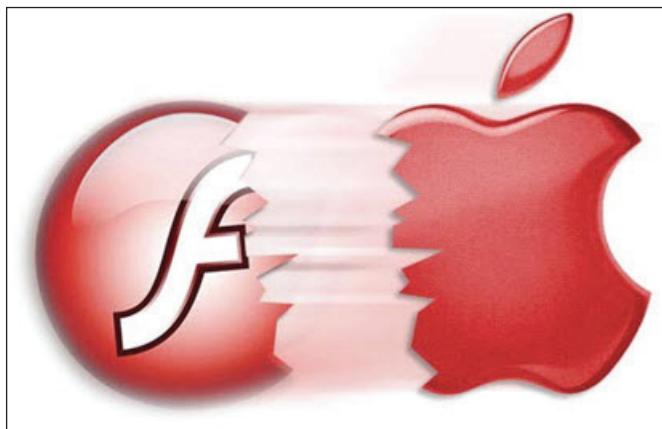


Mais après l'introduction du GIF, l'animation Web est restée pratiquement la même pendant encore 10 ans. À la fin des années 90, avec l'introduction d'une technologie appelée Flash, tout sur le Web s'est animé.

Flash est une plate-forme logicielle multimédia désormais obsolète créée par Adobe, utilisée pour produire des animations Web, des applications Internet riches, des applications de bureau, des applications mobiles et des jeux, et avec peu d'autres options pour créer des expériences attrayantes à l'époque, Flash a rapidement gagné en popularité parmi les développeurs et peuplé le Web pendant de nombreuses années.

Mais ensuite, le mobile est né, et avec lui, Flash a commencé à mourir parce que les appareils mobiles ne pouvaient pas gérer la charge CPU intensive que les fichiers Flash produisent lors de leur traitement, ce qui vide très rapidement la batterie d'un appareil. Mais non seulement l'éclosion des appareils mobiles a dicté l'avenir de Flash, mais Apple a décidé de ne pas prendre en charge Flash du tout dans aucun de leurs appareils mobiles, comme l'iPhone et l'iPad, améliorant ainsi considérablement les performances et la durée de vie de la batterie.

Après une bataille colossale entre Apple et Adobe, Apple a finalement remporté la victoire. Les utilisateurs ont commencé à désinstaller le lecteur Flash et les développeurs du monde entier ont commencé à supprimer définitivement tout le code Flash. La guerre entre ces deux sociétés géantes est ce qui a dicté la filière technologique utilisée pour réaliser les animations web de nos jours.



Mais, après la mort de Flash, Internet est redevenu un lieu statique, non animé ou semi-animé avec des GIF, jusqu'à la naissance du CSS, et plus précisément jusqu'à ce que CSS3 soit introduit. Donc, si le GIF était la première version de l'animation Web, elle a été suivie par la deuxième version appelée Flash, et elle a été suivie par CSS3 en tant que troisième version, mais JavaScript est devenu de plus en plus la technologie de choix au cours des dernières années, en particulier pour réaliser des animations Web volumineuses et complexes appelées séquences. Il convient de mentionner que l'introduction de HTML5 a également contribué à améliorer la façon dont l'animation Web se fait aujourd'hui.



Il existe une entière panoplie de frameworks JavaScript qui peuvent faire ce que CSS et JavaScript natif ne peuvent pas faire à eux seuls, la plus courante de ces bibliothèques est la plate-forme d'animation GreenSock ou GSAP, une bibliothèque d'animation conçue à l'origine pour ActionScript (une programmation orientée objet utilisée principalement pour le développement de sites Web ciblant la plate-forme Adobe Flash Player sous forme de fichiers SWF intégrés), qui a ensuite été réécrit en JavaScript.

Et il y a aussi la nouvelle API Web Animations, une technologie toujours en développement, qui promet une prise en charge native du navigateur pour réaliser des séquences d'animation complexes. Cette technologie est celle que nous allons voir plus tard dans le cadre de ce cours lorsque nous ferons des animations avec JavaScript.

Aujourd'hui, les mouvements sur les sites Web sont essentiellement réalisés à l'aide de transitions CSS, de keyframes CSS, de JavaScript ou d'une combinaison de ces éléments. Donc, si vous voyez une animation impressionnante sur un site Web, une annonce Web ou tout autre produit Web, il y a de fortes chances que vous regardiez une animation créée avec l'une de ces trois technologies. Nous allons commencer le cours avec les techniques offertes par CSS et quelques plug-in pratiques. Ensuite nous irons plus en profondeur avec le JavaScript. Nous y verrons l'utilisation des Sprites et nous terminerons en beauté avec un jeu utilisant l'API Web animations et son canvas.

Animations Web avec HTML et CSS

Il y a de nombreuses années, les sites Web se concentraient davantage sur l’affichage d’informations aux utilisateurs sans se demander comment créer des expériences visuelles pour rendre le site plus convivial. Au cours des dernières années, beaucoup de choses ont changé: les propriétaires de sites Web créent maintenant des expériences visuelles pour garder les utilisateurs sur leur site.

Les développeurs ont découvert que les êtres humains ont tendance à accorder plus d’attention aux objets en mouvement en raison de notre réflexe naturel de remarquer le mouvement.

Par extension, l’ajout d’animations à votre site Web ou à votre application est une méthode très importante pour attirer l’attention des utilisateurs sur des zones importantes de votre site Web et révéler plus d’informations sur un produit.

L’animation Web fait bouger les choses sur le Web. Elle est nécessaire pour créer des sites Web accrocheurs qui permettent de meilleures conversions et incitent les utilisateurs à cliquer, afficher et acheter des éléments sur votre site Web.

Nous l’avons déjà mentionné, mais lorsqu’elles sont bien faites, les animations peuvent ajouter une interaction précieuse, améliorer l’expérience émotionnelle des utilisateurs et ajouter de la personnalité à votre interface.

À l’heure actuelle, il existe des centaines de bibliothèques, d’outils et de plugins qui peuvent être utilisés pour créer des animations allant du simple au complexe. Avec CSS Animation, il devient inutile d’utiliser des plugins qui ralentissent la vitesse de votre site Web pour des animations qui peuvent être réalisées facilement avec CSS.

Récapitulation

- Aux débuts d’Internet, les pages web étaient de simples documents, constitués de texte et de liens vers d’autres pages.
- Le cerveau humain est conçu pour remarquer les mouvements.
- Les techniques d’animation web n’ont cessé d’évoluer depuis les débuts du web : GIF, Flash, etc.
- Aujourd’hui, il existe de nombreuses techniques d’animation web : JavaScript, SVG, Canvas, WebGL mais surtout CSS.
- L’animation permet d’ajouter de la vie à une page web, et ainsi de vraiment impacter l’expérience d’un utilisateur sur cette page.

Notes

[illegible]



Leçon 1-B



Les 12 principes de l'animation

Pour produire les films cultes que sont Blanche-Neige et les sept nains ou Cendrillon, Disney employait un noyau dur d'animateurs, responsables du style et des techniques, qui ont fait de ces films des monuments de l'animation. Ils sont rentrés dans l'histoire sous le nom de *Neuf Sages*.

Parmi les neuf, Ollie Johnston et Frank Thomas ont coécrit un livre intitulé *The Illusion of Life*. Ils y détaillent leurs méthodes pour donner vie à leur création par ce qu'ils appellent les "12 principes de l'animation". Bien que ces règles aient été établies dans le cadre de l'animation à la main traditionnelle, elles ont depuis été adaptées à pratiquement tous les autres styles d'animation, et restent parfaitement pertinentes dans le domaine de l'animation CSS.

L'animation peut expliquer tout ce que l'esprit de l'homme peut concevoir.

OBJECTIFS

À la fin de ce premier chapitre, vous serez en mesure :

- d'énumérer les 12 principes de l'animation Web.

Compression et étirement

Ce principe permet de donner de la masse à un objet, en l'étirant quand il accélère, puis en l'aplatissant et en l'élargissant quand il ralentit. Imaginez une balle en caoutchouc qui rebondit sur le sol en s'aplatissant avant de rebondir.

Anticipation

Le principe de l'anticipation permet de préparer l'utilisateur à une action à venir, lui donnant ainsi plus d'impact. Un chat qui se trémousse avant de bondir, le bras d'Astérix qui tourne avant de décocher un coup à un Romain... Les personnages animés ont besoin de mouvements préliminaires pour donner de la force à leurs actions. À l'inverse, un mouvement sans anticipation paraîtra lisse et artificiel.

Mise en scène

La mise en scène, dans le contexte de l'animation, désigne la composition. Il s'agit de guider les yeux des utilisateurs, vers les parties importantes de l'écran par l'utilisation de mouvements. Par exemple, cela peut se réaliser en faisant ressortir le bouton sur lequel nous voulons que l'utilisateur clique.

Straight Ahead et Pose to Pose (toute l'action d'un coup et partie par partie)

Straight Ahead et Pose to Pose sont deux différentes techniques qui permettent de créer des animations traditionnelles qui ne s'appliquent pas vraiment à l'animation web. Cependant, en CSS, il existe aussi deux différentes méthodes d'animation, les transitions et les keyframes, que nous étudierons au prochain chapitre. Les transitions sont basées sur une valeur de début et une valeur de fin, entre lesquelles le navigateur crée l'animation. Avec les keyframes, nous apprendrons à créer des animations par étapes, partie par partie.

Continuité du mouvement initial et chevauchement de deux mouvements consécutifs

Dans le monde réel, la masse et la densité des objets ne sont pas uniformes, ils accélèrent et décélèrent donc à des vitesses différentes. Imaginez que vous êtes dans une voiture et que votre tête est plaquée à l'appuie-tête lors de l'accélération, ou qu'elle est balancée vers l'avant quand le conducteur pile. Sans action de continuité ou de chevauchement, vous resteriez statique dans la voiture lors des changements de vitesse. Le fait de donner des vitesses différentes aux différentes parties d'un objet ajoute de l'authenticité à nos animations, et contribue à lui donner une masse réaliste.

Ralentissement en début et en fin de mouvement

Aussi connu comme *ease-in* et *ease-out*, que vous découvrirez plus loin dans ce cours, ce principe est basé sur le fait que les objets ne démarrent pas et ne s'arrêtent pas instantanément. Ils accélèrent et décélèrent. En appliquant des valeurs différentes à nos accélérations et décélérations, on peut modifier la perception de la taille ou de la masse d'un objet, rendre les mouvements plus vifs, ou plus laborieux pour un objet lourd.

Arc

La nature ne crée pas de lignes droites, et les mouvements naturels ne font pas exception à la règle. Cet exemple est un peu plus compliqué à exprimer avec une boîte en animation CSS, mais l'introduction d'arcs dans le mouvement d'un objet peut aider à le rendre plus naturel.

Action secondaire

Ajouter des animations pour séparer différentes parties d'une scène peut aider à accentuer ou renforcer les principaux éléments d'une animation.

Timing

Les objets doivent se déplacer à des vitesses crédibles par rapport à leur taille et à leur masse. Un semi-remorque accélère et décélère plus lentement et de manière plus homogène qu'une Ferrari, qui peut accélérer en un clin d'œil et s'arrêter quasiment sur place. On contrôle le timing des objets par rapport à leurs tailles relatives, ainsi que par la durée de l'animation.

Exagération

Même si le but est bien de rendre les animations réalistes, trop pousser ce réalisme peut résulter en un mouvement austère et fade. Aller un peu au-delà des limites naturelles permet de donner un peu de caractère et de personnalité à une animation.

Notion de volume

Traditionnellement, ce principe fait référence à la façon de réaliser une scène en représentant correctement la perspective. Pour nous, ce principe concerne la structure du code de l'animation. Il est très important de comprendre les tenants et les aboutissants des propriétés qu'on utilise, pour s'assurer que les animations correspondent au résultat souhaité.

Charisme

Pour rendre nos animations plus dynamiques et plus intéressantes, on peut y ajouter quelques effets supplémentaires. Le centre de l'attention l'est parce qu'il attire tous les regards. C'est l'attrait et le charisme qui attirent l'attention, et la conservent.

Récapitulation

Les 12 principes de l'animation sont :

- Compression et étirement, pour compresser et étirer un élément ;
- Anticipation, pour préparer l'audience à un mouvement à venir ;
- Mise en scène, pour guider les yeux de l'utilisateur vers les éléments importants d'une page ;
- Straight Ahead et Pose to Pose (toute l'action d'un coup et partie par partie), qui correspond davantage à l'animation traditionnelle mais fait penser à la différence transitions/keyframes ;
- Continuité du mouvement initial et chevauchement de deux mouvements consécutifs, pour faire accélérer et décélérer un élément en fonction de sa masse ;
- Ralentissement en début et en fin de mouvement, basé sur le fait que les objets ne démarrent pas et ne s'arrêtent pas instantanément ;
- Arc, pour créer des mouvements naturels ;
- Action secondaire, pour séparer différentes parties d'une scène dans une animation ;
- Timing, pour faire se déplacer les objets à une vitesse crédible ;
- Exagération, pour donner du caractère et de la personnalité à une animation ;
- Notion de volume, pour que les animations correspondent au résultat souhaité ;
- Charisme, pour rendre les animations plus dynamiques.



Leçon 1-B

Utiliser un fond d'écran vidéo

Les premiers éléments qu'un visiteur voit sur votre site sont les plus importants, ce sont ces éléments qui détermineront si votre site répond au besoin du visiteur et qu'ils lui donnent envie de rester.

Dès son arrivée, vous avez que quelques secondes pour convaincre le visiteur de rester sur votre site web.

Pour éviter que ce visiteur ne quitte trop vite votre site internet, vous pouvez lui créer un chemin d'éléments qui apparaissent lorsqu'il fait défiler la page de votre site ou bien vous pouvez captiver son attention en lui montrant une vidéo dès la page d'accueil de votre site.

Dans cette première leçon, c'est cette seconde approche que nous vous expliquerons pour créer une vidéo en fond d'écran pour accueillir les internautes sur votre site.

OBJECTIFS

À la fin de ce premier chapitre, vous serez en mesure :

- de créer une page d'accueil avec vidéo et lien vers le site.

Créer la structure en HTML

Avant de modifier l'apparence de la vidéo, vous allez devoir créer la structure de l'élément en fond d'écran. Dans notre exemple, la structure HTML de la vidéo en fond sera composée d'une balise permettant d'intégrer la vidéo ainsi que de textes et contrôles superposés à la vidéo. À l'aide de Visual Studio Code, ouvrez le dossier **Page_Acces** que vous a remis votre instructeur et ajoutez le code suivant dans la page **index.html** que contient le dossier.

```
<!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>Exemple page d'accès</title>
  <link rel="stylesheet" href="Style/style.css">
</head>
<body>
  <video id="background-video" autoplay loop muted>
    <source src="Videos/river.mp4" type="video/mp4">
  </video>
  <div class="content">
    <h1>BIEN PLUS QU'UNE RIVIÈRE</h1>
    <h2>Une source de vie</h2>
    <a href="Accueil.html">Accéder au site</a>
  </div>
</body>
</html>
```

Vous devriez voir la vidéo s'afficher dans sa taille réelle suivie des deux éléments de titre **<h1>** et **<h2>** et d'un lien, élément **<a>**, comme le montre la prochaine figure.



BIEN PLUS QU'UNE RIVIÈRE

Une source de vie

[Accéder au site](#)

Créer le fichier css

Maintenant que les éléments sont créés, il ne reste plus qu'à les mettre en forme avec du code CSS. Dans le fichier *style.css* que contient les fichiers de votre site, ajoutez le code suivant afin de positionner votre vidéo en plein écran.

```
#background-video{
  height: 100vh;
  width: 100vw;
  object-fit: cover;
  position: fixed;
  left: 0;
  right: 0;
  top: 0;
  bottom: 0;
  z-index: -1;
}
```

Enregistrez vos modifications puis affichez votre page à l'écran. Votre vidéo s'affiche maintenant en utilisant tout l'espace disponible et tous les autres éléments sont affichés par-dessus cette dernière dans le coin supérieur gauche de l'écran, comme le montre la prochaine figure.



Ceci étant fait, nous allons ajouter le code css afin de formater les éléments de titre ainsi que le lien d'accès. Ajoutez le code suivant dans votre fichier *style.css*.

```
h1, h2, a{
  color: white;
  font-family: Trebuchet MS;
  font-weight: bold;
  text-align: center;
}

h1{
  font-size: 6rem;
  margin-top: 30vh;
}

h2{
  font-size: 3rem;
}
```

Dans ce dernier bloc de code, nous avons défini la couleur du texte, la police de caractères et le style de caractères ainsi que l'alignement du texte de chacun des éléments. Enregistrez vos modifications puis affichez votre page dans votre navigateur. Vous obtiendrez le résultat suivant affiché dans votre navigateur.



Bon, nos deux éléments de titre sont bien centrés sur notre page sauf notre lien. Un lien étant un élément de type *inline*, contrairement à nos deux éléments de titre qui sont des éléments de type *block*, ne peut pas être centré sur notre page en utilisant la propriété css *text-align*. Il faudra donc que nous utilisions une autre technique afin de centrer notre lien sous notre élément `<h2>`, comme le montre le prochain bloc de code. Ajoutez ce bloc de code à la suite du code déjà saisi dans votre fichier style.css.

```
a{
  font-size: 2rem;
  text-decoration: none;
  margin-left: 44%;
}
```

Ce dernier bloc de code modifiera la taille des caractères de notre lien afin de l'afficher un peu plus gros, supprimera le trait de soulignement affiché par défaut sous le lien et ajoutera un espace entre la bordure gauche de votre fenêtre et le texte du lien. Enregistrez vos modifications puis, affichez votre page dans votre navigateur. Vous devriez obtenir un résultat comme celui de la prochaine figure affiché à l'écran.



Création de notre page Accueil.html

Notre page d'accès est maintenant complétée et son apparence est bonne. Si vous cliquez sur le lien Accéder au site, vous obtiendrez une page blanche affichée à l'écran. Modifiez le fichier Accueil.html comme le montre le prochain bloc de code.

```
<!DOCTYPE html>
<html lang="fr">
<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>Accueil</title>
  <link rel="stylesheet" href="Style/style.css">
</head>
<body>
  <h1 id="accueil">Bienvenue sur le site Source de vie.</h1>
  <h2></h2>
</body>
</html>
```

Notre élément de titre `<h1>` prendra les mêmes paramètres d'affichage que l'élément `<h1>` de notre page d'accès soit, la couleur du texte, la même police de caractères de la même grosseur etc. Un texte affiché en blanc sur un fond blanc n'est pas ce que nous désirons voir sur notre page. Nous allons donc modifier apporter quelques modifications dans notre fichier style.css afin de s'assurer que le texte de notre élément `<h1>` de notre page s'affiche en utilisant une couleur et une police de caractères différentes de celle de notre page d'accès. Ajouter le bloc de code suivant à la fin de votre fichier style.css.

```
#accueil{
  color: navy;
  font-family: "Comic Sans MS";
  font-weight: bold;
  text-align: center;
  font-size: 3rem;
  margin-top: 20vh;
}
```

Maintenant, si vous cliquez sur le lien affiché sur votre page d'accès, vous serez redirigé vers la page Accueil.html qui affichera le contenu suivant dans votre navigateur.

Bienvenue sur le site Source de vie.



Récapitulation

Dans cette première partie du cours, nous nous sommes concentrés sur la création d'une page d'accès utilisant une vidéo afin d'attirer la curiosité de l'internaute. Ajouter une vidéo en plein écran est facile et rapide à mettre en place. Cet élément ne demande pas trop de connaissances techniques et il est possible d'avoir un rendu attractif et professionnel simplement grâce à ce fond vidéo.

Dans les prochains chapitres, nous utiliserons des techniques différentes afin de créer le même type de page d'accès pour nos sites Web.

[illegible]