

# Animations

Swift UI

# Transformation

Il faut faire une transformation sur l'élément en question.

Exemple :

- .rotation3DEffect()
- .shadow()
- .scaleEffect()
- ...

# Animation

Il faut ensuite ajouter une animation :

- `.animation()`

L'animation sert à effectuer les transformations listées au dessus  
(L'ordre dans les modifier est à prendre en compte)

Il est possible de mettre l'animation à `none` pour ne pas avoir d'animation sur une transformation

Pour créer les animation il est nécessaire d'avoir une variable Boolean pour préciser quand effectuer l'animation.

# Fonctionnement

En codant des animations comme expliqué précédemment, nous changeons l'état de notre vue. Swift UI lui va calculer la différence entre l'état initial et le nouvel état et va créer des conteneurs d'animation pour les vue possédant un changement. Il va utiliser des animateurs afin de d'animer la transformation nécessaire pour passer de l'état initial à l'état final. Petit à petit Swift UI va modifier les propriétés animés pour obtenir une animation.