

# Application Staves

Cette application permet de faire des animations autour des notes de musique, à fin pédagogique.

- [Animation](#)
- [Les notes](#)
- [Les accords](#)
- [Les portées](#)

## Création d'une animation

---

### Table des matières

- [Introduction](#)
- [Composer le code de l'animation](#)
  - [Commentaires dans le code](#)
  - [Faire une pause](#)
  - [Avancer sur la portée](#)
  - [Écrire un texte général](#)
  - ["Nettoyer" l'animation \(tout effacer\)](#)

### Introduction

Une animation est composée de `pas` (step). Chaque pas exécute une action. Ces pas se définissent dans la console (ie dans le code) en passant à la ligne. **Chaque ligne représente donc un pas.**

Ces pas peuvent : \* Construire une note (et l'afficher) \* Construire une portée \* Écrire un texte associé à un objet (note, portée, barre, etc.) \* Déplacer un objet quelconque (surtout les notes, les portées ne se déplacent pas)

### Composer le code de l'animation

## Écrire un commentaire

Les commentaires s'écrivent avec `#` en début de ligne.

Noter que les commentaires ne peuvent pas se trouver sur un pas proprement, même après le code. Il faut absolument qu'ils soient sur une ligne seule, qui sera passée.

## Faire une pause dans l'animation.

On utilise la méthode `WAIT` pour faire une pause, avec en argument le nombre de secondes (qui peut être un flottant, pour des temps très courts) :

```
WAIT(<nombre de secondes>)
```

Par exemple, pour attendre 4 secondes :

```
maNote=NOTE(a3)
WAIT(4)           // 4 secondes avant de construire l
'accord
monAccord('c3 eb3 g3')
```

*Note : C'est un "pas" comme les autres, donc il doit être mis sur une ligne seule comme toute étape.*

## Se déplacer sur la portée

La commande pour écrire à la suite des dernières notes sur la portée, on utilise la commande :

```
LEFT([<nombre pixels>])
```

Par défaut (sans argument), le déplacement sera de 40px (`Anim.default.hoffset`). Sinon, le déplacement sera de la valeur précisée.

Par exemple :

```
LEFT()  
// => les notes suivantes s'éciront 40px plus à gauche  
  
LEFT(100)  
// => les notes suivantes s'éciront 100px plus à gauche
```

## Écrire un texte général

Pour écrire un texte en rapport avec l'animation (en haut à gauche), on utilise la méthode `WRITE` :

`WRITE()`

Par exemple :

`WRITE("Ce qu'il faut remarquer à ce moment-là.")`

*Noter que les objets tels que les notes ont également leur propre méthode textuelle, qui permet d'aligner le textes directement à ces notes, accords, etc.*

## Nettoyer l'animation

“Nettoyer l'animation” signifie supprimer tous les éléments affichés. On peut ou non demander que les portées restent en place, though.

Le pas à utiliser est :

```
CLEAR(<avec les portées>)
```

... `<avec les portées>` est `FALSE` par défaut, donc il faut ajouter `true` pour effacer aussi les portées :

```
CLEAR(true) // efface aussi les portées
```

---

## Les Notes

### Table des matières

[Désignation des notes](#) [Constantes notes](#) [Créer une note](#) [Déplacer une note](#)  
[Placer une note sur une portée précise](#) [Détruire d'une note](#)

## Désignation des notes

Les notes doivent être désignées par :

```
<nom note><alteration><octave>
```

- On désigne le `nom des notes` par une seule lettre, anglaise, de "a" (la) à "g" (sol).
- L'altération est soit rien (note naturelle), soit un signe (cf. ci-dessous "b", "d", "x", ou "t").
- Vient ensuite l'octave, un nombre, négatif si nécessaire (*mais pour le moment, on va seulement jusqu'à l'octave 0, les autres ne sont pas gérés*).

## Marque des altérations

La valeur `<alteration>` ci-dessus peut être :

- `b` pour bémol
- `t` (comme "ton") pour double bémol
- `d` pour dièse
- `x` pour double-dièse

## Constantes notes

De l'octave 0 (qui n'existe pas en français) à l'octave 7 il existe des constantes pour chaque note avec l'altération bémol ("b") et dièse ("d").

Les deux formules suivantes sont donc possible :

```
no=NOTE(ad5) no=NOTE('ad5')
```

Note: Attention à ne pas donner à une variable note le nom d'une de ces constantes. Par exemple, si on fait :

```
a5=NOTE ( a5 )
```

Cela génèrera une erreur de constante déjà définie.

On peut utiliser plutôt :

```
na5=NOTE(a5)
```

## Créer une note

```
<variable name>=NOTE(<note>)
```

- `<variable>` peut avoir le nom qu'on veut, HORMIS un nom de constante, comme `a5`.
- La `<note>` peut être soit un string soit une constante (cf. [Désignation des notes](#))
- Une telle séquence (un pas) ne doit pas comporter d'espaces.

## Déplacer une note

Pour déplacer une note, on utilise le pas :

```
<nom variable note>.moveTo(<nouvelle note>[,<params>])
```

Exemple :

```
maNote=NOTE(a4) // crée la note LA 4  
maNote.moveTo(g4) // descend la note vers SOL4
```

*Ne mettre aucune espace dans ce code.*

*Noter que la note de destination devra vraiment la nouvelle valeur de la note. Si la note "a4" se déplace vers "a3", cette note deviendra "a3" dans ses données.*

## Placer une note sur une portée précise

Si on veut placer une note sur une portée hors de la portée active, on indique l'indice de cette portée avant la note, puis ":" :

```
<indice portée>:<note>
```

Par exemple :

```
note_autre_staff=NOTE('2:a4')  
OU  
note_autre_staff=NOTE('2:'+a4)
```

... placera un LA4 sur la deuxième portée, même si elle n'est pas active.

*Noter que cela ne rend pas la portée active.*

## Détruire d'une note

Pour détruire la note (la supprimer de l'affichage, utiliser :

```
<nom variable note>.remove()
```

////////////////////////////////////

# Les Accords

---

## Table des matières

[Création d'un accord](#) [Référence aux notes de l'accord](#) \* [Destruction d'un accord](#)

## Création d'un accord

On crée un accord avec :

```
monAccord=CHORD(' <note1> <note2>... <noteN>')
```

... où chaque note doit correspondre à la définition normale.

Par exemple :

```
accDom=CHORD('c3 eb3 g3')
```

## Référence aux notes de l'accord

On fait appel aux notes de l'accord par :

```
<nom accord>[<indice note>]
```

... où `<indice note>` est l'indice 1-start.

Par exemple :

```
accDom=CHORD('c3 eb3 g3')  
accDom[1].moveTo('c4')  
// Prends la première note (c3) et la déplace en c4.
```

## Destruction d'un accord

Pour détruire l'accord, utiliser :

```
<nom variable accord>.remove()
```

# Les portées

---

## Table des matières

[Créer une portée](#) [Activer une portée](#)

## Créer une portée

Pour afficher une portée, utiliser le pas :

```
NEW_STAFF(<clé>)
```

... où `<cle>` peut être `SOL` , `FA` , `UT3` , `UT4` .

Par exemple :

```
NEW_STAFF(SOL)
```

... qui affichera une portée en clé de sol en dessous de la dernière portée.

*Noter que cette portée deviendra la portée active, c'est-à-dire celle où seront placées les objets définis par la suite.*

## Activer une portée

Activer une portée signifie que tous les pas suivants la viseront. Par exemple, les notes se déposent toujours sur la portée active.

```
ACTIVE_STAFF(<indice de la portee>)
```

... où `<indice de la portee>` est son rang dans l'affichage, en partant de 1 et du haut. Donc la portée la plus en haut s'active par :

```
ACTIVE_STAFF(1)
```