

# MuScaT

## Manuel d'utilisation

### Introduction (histoire)

Suite à la diffusion de mon Initiation à l'analyse musicale — et son « succès » que je n'aurais jamais imaginé aussi grand — nombreux ont été les professeurs et autres pédagogues ou musicologues à me demander le logiciel que j'avais utilisé pour en créer l'animation.

C'est malheureusement une application personnelle un peu trop... personnelle (comprendre : indomptable pour qui ne l'a pas créé), une usine à gaz ne fonctionnant qu'à la ligne de code (son créateur ne sait même pas toujours par quel bout la prendre).

Mais pour répondre à ces marques d'intérêts et à mes propres besoin, j'ai voulu concevoir un outil plus simple et plus pratique qui permettrait de réaliser rapidement des analyses de partitions (entendu que c'est toujours un peu fastidieux et que le résultat manque souvent d'homogénéité).

C'est ainsi qu'est née l'idée de **MuScaT** — dont le nom est composé de « Mu » pour « Musique », « Sc » pour « Score » (« partition » en anglais) et « Ta » à l'envers pour « Tag », le sens en français, comme les tags qu'on *taggue* sur les murs.

En bref, **MuScaT** permet de **réaliser rapidement, de façon très propre et très pratique, des analyses de partitions musicales** comme l'extrait ci-dessous.

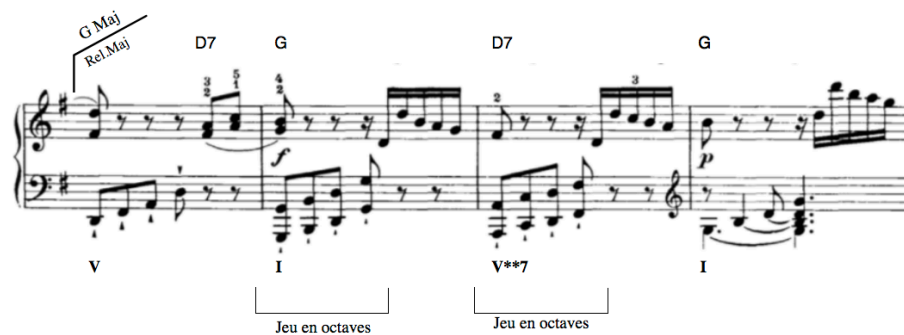


Figure 1: Extrait d'analyse musicale avec MuScaT

Elle est semi-graphique, et permet d'ajuster très finement les TAGs — au pixel près — de façon visuelle et agréable.

- Synopsis général de fabrication
- Composition d'un tag
  - Verrouillage des tags
- Les Images
  - Définition de la taille d'une image
  - Séquence d'images
- Tous les types (natures) d'éléments
  - Second mot (contenu, accord)
  - Autres données de la ligne
  - Les types de textes
- Les Options
- Les Utilitaires
  - Changement du dossier des captures écran (Mac)
  - Renommage des fichiers images (Mac/Unix)
  - Création d'une nouvelle analyse (Mac)
  - Activation d'une analyse (Mac)

## Synopsis général de l'analyse

Commençons par un aperçu du processus général qui va permettre de produire une analyse musicale à l'aide de **MuScaT**.

1. Création du dossier de l'analyse,
2. découpage de la partition en « images-systèmes »,
3. inscription des images-systèmes dans l'analyse,
4. ajouts des accords, des chiffrages, des cadences, de tous les éléments d'analyse,
5. Positionnement des éléments graphiques,
6. récupération du code final,
7. impression en PDF.

## Synopsis détaillé de l'analyse

### Création du dossier de l'analyse

Si vous êtes sur Mac, le plus simple est d'utiliser le script `create.rb` qui fait tout le travail pour vous, simplement en lui donnant le nom de l'analyse.

```
> cd /chemin/vers/dossier/MuScaT
> ./utils/create.rb "Ma première analyse"
```

Si vous n'êtes pas sur Mac, la procédure est à peine plus compliquée : vous dupliquez le dossier `Template` qui se trouve dans le dossier `MuScaT/_analyses_` (ce dossier est le dossier qui contient toutes les analyses ; même si vous pouvez les ranger où vous voulez).



Appelons ce dossier `monAnalyse` pour illustrer.



Dans ce dossier, vous pouvez mettre, au même niveau que le dossier `analyse`, votre partition en PDF ou en image.

### Découpage de la partition en « images-systèmes »

Si la partition que l'on s'apprête à analyser est suffisamment aérée (espace entre les systèmes), on peut la garder telle qu'elle. Dans le cas contraire (et le plus fréquent), il faut découper cette partition en systèmes, c'est-à-dire faire une image de chaque système pour les écarter sur la table d'analyse.

Dans tous les cas, on place la ou les images dans le dossier `monAnalyse/analyse/images`.



Notez que c'est ce dossier `analyse` que vous déplacerez dans le dossier `_table_analyse_` pour visualiser ou travailler votre analyse.

### Inscription des images-systèmes

On ouvre ensuite son fichier `monAnalyse/analyse/tags.js`. C'est le **fichier principal de l'analyse**, celui qui va définir tous les éléments, les images, les

```
38
39
40 Tags = `
41 # Définir ci-dessous vos tags (aide: [TODO])
42 # -----
43
44 # Premier système
45 # -----
46
47 score ligne-01.png x=2 y=100
48
49 # Deuxième système
50 # -----
51
52 score ligne-02.png x=2 y=203
53
54 # Troisième système
55 # -----
56
57 score ligne-03.png x=2 y=269
58
59 `;
60
61
62
63
64
65
66
67
68
```

Figure 2: Exemple d'images dans tags.js

marques de modulations, les accords, les cadences, les parties, tout ce qui constitue l'analyse.

Vous devez ouvrir ce fichier en texte simple, c'est-à-dire ne surtout pas utiliser de traitement de texte, ou alors en s'assurant d'exporter le fichier final au format « SimpleText » (.txt).

Dans ce fichier `tags.js` On définit d'abord les images de la partition, en ajoutant des commentaires pour pouvoir se retrouver, plus tard, lorsque le fichier deviendra conséquent. Par exemple :

```
// Dans tags.js
option('code');

Tags = `
// Premier système, les mesures de 1 à 10
partition system-1-mes-1-10.png

// Deuxième système, les mesures de 11 à 16
partition system-2-mes-11-16.png

// Troisième système
// ... etc.

`;
```

Note : l'option 'code', en haut du fichier `tags.js`, permet simplement de voir le code à côté de la table d'analyse.

## Définition de tous les éléments de l'analyse

L'élément graphique de base de l'application **MuScaT** est le « TAG » (comme on en parle sur les murs des villes). Une analyse avec **MuScaT** consiste donc à « tagguer » une partition (remarquez que les partitions elles-mêmes, ou les images de leurs systèmes, sont aussi des « TAGs »). C'est la raison pour laquelle le fichier qui va les définir s'appelle `tags.js`.

On définit tous les autres éléments graphiques, tous les *tags* (cf. pour le détail de la procédure, voir Composition d'un tag) : marque de parties, accords, chiffrages, numéros de portée, cadences, textes divers, etc. Le mieux est de s'arranger pour les placer, dans `tags.js`, à peu près en fonction des positions dans l'analyse. Si une cadence doit se produire sur le troisième système, il vaut mieux la définir après la ligne insérant l'image de ce troisième système (remarquez cependant qu'il n'y a aucune obligation là-dessus, vous pouvez aussi, rassembler tous les accords d'un côté, toutes les cadences de l'autre, etc. à votre guise).

## Positionnement des éléments graphiques

Pour activer cette nouvelle analyse, nous allons placer son dossier **analyse** dans le dossier **\_table\_analyse\_** de **MuScaT**.

Pour se faire, il faudra certainement détruire le dossier **analyse** qui se trouve déjà là. C'est la raison pour laquelle il est vivement conseillé de travailler avec des duplicatas plutôt qu'avec les dossiers d'analyse originaux.

Les heureux possesseurs de Mac ont un script qui permet d'activer très simplement n'importe quelle analyse. Il suffit de jouer `./utils/analyse.rb` lorsqu'on se trouve dans le dossier de **MuScaT** ou simplement `analyse.rb` si on se trouve dans le dossier `MuScaT/utils/`.

Pour les autres, il faut donc faire une duplication du dossier **analyse**, la placer dans le dossier **\_table\_analyse\_** et enfin ouvrir le fichier **\_table\_analyse\_/partition.html** dans un navigateur internet pour travailler l'analyse (Firefox est le meilleur choix, pour **MuScaT**).

On peut placer les éléments aux bons endroits simplement en les déplaçant à la souris, ou avec les flèches de son clavier. On peut en ajouter des nouveaux en dupliquant les lignes de code ou les ajoutant explicitement dans le code.

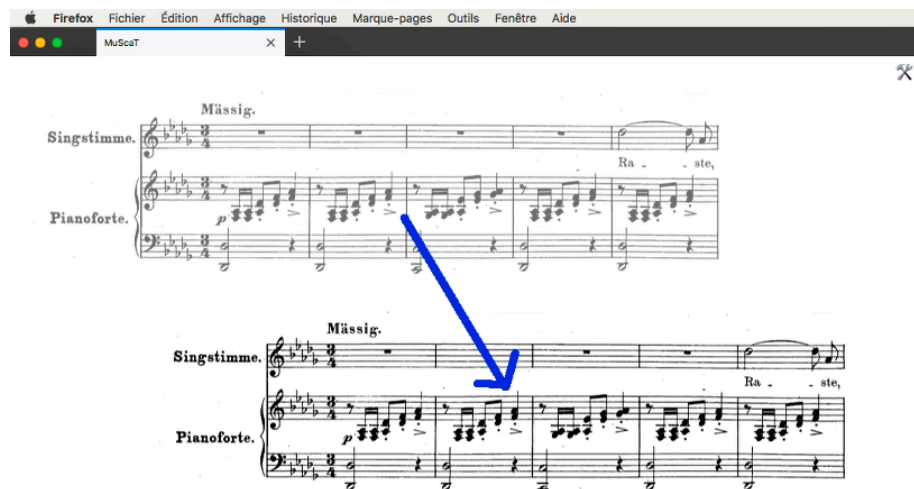


Figure 3: Exemple de déplacement d'élément

Sans l'option `option('code')` activée, il faut modifier le code directement dans le fichier `tags.js` puis recharger la page dans Firefox.

## Lignes repères

Pour faciliter l’alignement des TAGs — par exemple l’alignement des dernières mesures de fin des systèmes — on peut utiliser des lignes de repère. Pour cela, il suffit d’activer l’option **repères** (ou **reperes** ou **lines of reference**).

Cela ajoute deux lignes à l’écran, une verticale et une horizontale, qu’on peut déplacer à loisir.

## Récupération du code final

Si l’on a travaillé dans le champ de texte à côté de la table d’analyse, on doit copier le code final dans le fichier **tags.js**, au risque de perdre tous les changements. Pour se faire, on clique sur le bouton des outils — en haut à gauche — et on demande à mettre le code complet dans le presse-papier. On colle ce code dans le fichier **tags.js**, en remplaçant l’intégralité de son contenu.

On n’oublie pas de faire une duplication dans l’autre sens, vers le dossier **\_analyses\_/monAnalyse/** pour conserver les modifications opérées.

## Impression de l’analyse en PDF

On imprime la page HTML du navigateur en choisissant le format PDF (ou on enregistre la page au format HTML et on utilise un outil de transformation des pages HTML en PDF).

## Et voilà

Et voilà, c’est fait ! Et vous pourrez retoucher à votre analyse à n’importe quel moment grâce au dossier **analyse** qui contient tous les éléments de l’analyse et qu’il suffit de mettre sur la table d’analyse (comprendre : dans le dossier **\_table\_analyse\_**).

## Composition d’un tag

Voyons plus en détail comment se compose une ligne du fichier **tags.js**, une ligne définissant un *tag* ou une partition.

Un *TAG* — image de la partition comprise — se compose d’une ligne dans le fichier de données.

Cette ligne a le format général suivant :

```
<nature>[ <contenu>][ <coordonnées>][ <options, type>]
```

Par exemple, pour une cadence (nature = 'cadence') de « V I » (contenu = 'V\_I') qu'on veut placer à 200 pixels depuis le haut (coordonnée y = 200) et 100 pixels de la gauche (coordonnées x = 100), de type « cadence parfaite » (type = 'parfaite'), on insèrera dans son fichier `tags.js`, sous la définition de l'image (« score ») :

```
Tags = `

    score ma_partition.jpg y=100 x=10

    cadence V_I type=parfaite y=200 x=100

    modulation G_min x=200 y=100

`;
```

Une « nature » de TAG (le premier mot), peut toujours être exprimé par ses trois premières lettres (exception faite du terme « partition » qui rentrerait en conflit avec « partie »). Ainsi, on peut écrire le code ci-dessous :

```
Tags = `

    sco ma_partition.jpg y=100 x=10

    cad V_I type=parfaite y=200 x=100

    mod G_min x=200 y=100

`;
```

L'intégralité des natures d'éléments est détaillé ici.

Vous observerez que tout de suite après la création, un identifiant est ajouté à toutes les lignes, mêmes les lignes vides. Il contient de ne pas y toucher, sous peine de voir son travail réduit à néant.

Ainsi, le code ci-dessous, au final, donnera :

```
# Contenu intégral du fichier tags.js
option('code'); // pour voir ce code à côté de la partition

Tags = `
    sco ma_partition.jpg id=2 y=100 x=10
    #3#
    cad V_I type=parfaite id=4 y=200 x=100
    #5#
```



```
mod G_min id=6 y=100 x=200
`;
```

### Forme raccourcie d'écriture

Pour la première définition du TAG, on peut utiliser une version raccourcie de définition qui la rend très simple et très rapide. Elle consiste à utiliser :

```
Tags = `
<version 3 lettres|normale> <contenu|source> <valeur x> <valeur y>
`;
```

Par exemple, pour une *modulation* vers la tonalité de SOL mineur (G min.) qui doit se situer à 200 pixels du haut et 450 pixels de la gauche, on pourra écrire simplement :

```
mod G_min 200 450
```

### Verrouillage des tags

On peut « verrouiller » un TAG, c'est-à-dire empêcher totalement ses modifications, aussi bien sa position que son contenu, en ajoutant un astérisque, un rond (ALT #) ou même un `!` au tout début de sa ligne (suivi ou non par une espace).

**MuScaT** ajoutera un vrai cadenas ( `!` ) qui rendra ce verrouillage très visuel.

Une fois verrouillé, le TAG ne peut plus être déplacé à la souris. En revanche, il peut tout à fait être modifiée dans le code (sa position, son contenu, etc) pour un ajustement très précis.

Pour deverrouiller un TAG et le rendre à nouveau mobile, il suffit tout simplement de retirer cette marque de verrouillage dans le code.

### Les Images

Il existe trois mots clés pour indiquer la nature d'une image, mais ils sont identiques en réalité : **image**, **score** ou **partition**. C'est le premier mot à trouver sur la ligne d'une image. Juste après, on doit trouver le nom de cette image, ou son chemin relatif depuis le dossier **image** du dossier de votre analyse.

```
partition haydn/premier_mouvement.png [...]
```

Ci-dessus, l'image **premier\_mouvement.png** doit donc se trouver dans le dossier `./analyse/images/haydn/` de votre dossier d'analyse.

## Définition de la taille d'une image

On peut définir la taille d'une image à l'aide du paramètre `w` (ou `width`, « largeur », en anglais). Sa valeur peut être explicite avec une unité, explicite sans unité ou en pourcentage. Par exemple :

```
// Dans tags.js
Tags = `
sco image-0.png
sco image-1.png w=200
sco image-2.png w=10cm
sco image-3.png w=50%
`;
```

Avec le code ci-dessus, l'image 0 aura sa taille normale, `image-1` fera 200 pixels de large, `image-2` fera 10 centimètres de large et `image-3` sera mise à 50% de sa largeur.

## Séquence d'images

Bien souvent, une analyse n'est pas constituée d'une seule image pour toute la partition. Il y a trop peu d'espace entre les systèmes. On conseille donc fortement de découper les partitions en autant de systèmes qu'elles en comportent (vous trouverez des indications sur la procédure de découpage de la partition ci-dessous).

Mais il serait fastidieux d'entrer la ligne de chaque image de système dans notre fichier `tags.js`. Une partition même courte peut très vite comporter de 10 à 15 systèmes et ce serait autant de lignes de partition qu'il faudrait introduire dans le code...

Au lieu de ça, si les images des systèmes ont été nommés en respectant une règle simple (avec des suites de nombres), une seule ligne suffira pour entrer tous les systèmes de la partition. Par exemple :

```
score mouvement_1/image-[1-35].png
```

Le texte ci-dessus indique qu'il y a 35 images de système dans ce mouvement. Le code qui en résultera sera :

```
score mouvement_1/image-1.png
score mouvement_1/image-2.png
score mouvement_1/image-3.png
score mouvement_1/image-4.png
...
...
```

```
score mouvement_1/image-35.png
```

Nous vous invitons vivement à commencer par cette opération avant l'insertion de toute autre marque sur la partition.

Quand **MuScaT** place les images sur la table d'analyse, il les répartit pour obtenir l'aspect initial de la partition. On peut modifier ce comportement en définissant explicitement un espace (vertical) entre chaque système ou chaque image, grâce à l'option `espacement images` :

```
// Code intégrale du fichier tags.js
option('code');option('espacement images', 50);

Tags=`
sco haydn/mouvement_1-[1-35].png
`;
```

Notez la version raccourci de la nature du TAG : `sco` pour `score`.

Notez également l'usage de l'option `code` qui permet d'afficher le code à côté de la table de l'analyse, pour pouvoir le modifier.

Grâce à l'option `espacement images` défini ci-dessus, chaque image (chaque système) sera séparé de 50 pixels.

Une fois ce code établi, vous pouvez déplacer les images dans la page pour les ajuster à vos besoins. Cela créera automatiquement les `x` et les `y` des coordonnées spatiales de chaque système au bout des lignes de score.

Astuce : si votre écran est assez grand et que vous adoptez l'option `code beside` (ou `code à côté`), vous pourrez voir en direct votre code s'actualiser.

---

## Nature des tags

Détaillons toutes les natures de TAGs qu'on peut trouver.

Dans la ligne, le premier mot définit la `<nature>` du tag.

Cette nature peut être (note : les deux mots, français et anglais, sont utilisables) :

```
partition    `image <source> x=... y=... z=...`
Exemple : `image monScore.png z=50 x=100 y=100`
- "z" désigne le zoom en pourcentage
- l'image doit se trouver dans le dossier 'images'
Alias : 'score', 'image'
Note : un astérisque ("*") indique au départ une
suite d'image (image1, image2, image3 etc.)
```

```

mesure      `mesure <nombre> x=... y=...`
mes         Exemple : `mes 13 x=100 y=234`
           Alias : 'measure'

accord      `accord <nom> x=... y=...`
acc         Exemple : `accord Cm7 x=230 y=520`
           Alias : 'chord'

harmonie    `harmonie <degré accord et renversement> x=... y=...`
har         Exemple : `harmonie II** x=200 y=230`
           Alias : 'harmony', 'chiffre'

modulation  `modulation <Ton[/sous-texte]> x=HH y=VV h=HH`
mod         Exemple : `modulation D_Maj/Sous-dom. x=100 y=100 h=60`
           « h », ici, permet de définir la longueur du trait qui
           rejoint la partition (le trait vertical).

cadence     `cadence <degré accord> type=<type cadence> x=... y=... w=...`
cad         Exemple : `cadence I type=italienne w=200 x=12 y=100`

ligne       `ligne <type ligne> x=... y=... w=...`
lig         Exemples : `ligne U w=120 x=100 y=50`
           `line |---| w=50 x=100 y=50`
           Alias : 'line'

degré       `degre <indice> x=... y=...`
deg         Exemple : `degre 5 x=100 y=120`
           Alias : 'degree'

texte       `texte <contenu> x=... y=... type=...`
tex         Exemple : `texte Exposition x=100 y=50 type=partie`
           Alias : 'text'

```

### Contenu du tag (second mot)

Le seconde « mot » définit le plus souvent le contenu textuel ou, pour les images, le nom du fichier dans le dossier **analyse/images**. C'est aussi, souvent, un accord ou son chiffre.

On peut par exemple écrire un texte quelconque à une position quelconque avec la ligne :

Tags = `

```

texte Et_si_j'étais_un_texte_quelconque x=300 y=400

```

`;

Remarquez comme les espaces ont été remplacées par des tirets plats (qu'on obtient sur Mac avec la combinaison de touches Maj- — touche majuscule et tiret).

Ce deuxième « mot » de la ligne sert aussi par exemple à définir le type des lignes à obtenir (cf. Dessiner des lignes).

### Autres données de la ligne

Les deux autres informations capitales sont les positions verticale et horizontale du tag à poser (ou de la partition).

NOTE IMPORTANTE : dans votre fichier **tags.js**, ces valeurs peuvent dans un premier temps être approximatives, et seront affinées directement à l'écran.

On définit position verticale avec **y=** et la position horizontale avec **x=**, comme nous l'avons vu dans les exemples précédents. Le nombre est exprimé en pixels.

Pour les lignes et les cadences par exemple, on peut définir aussi la largeur avec la lettre « w » qui signifie « width » (largeur) en anglais : **w=200**. Le nombre correspond là aussi au nombre de pixels.

Ensuite, on peut définir certaines choses comme le « type » du tag. On l'a vu pour la cadence, par exemple. Les autres tags pouvant définir leur type sont le **texte** ou la **ligne** (bien que la **ligne** se définit plutôt par son contenu).

### Écrire des textes

Ce que l'on appelle les « textes », ici, ce sont tous les textes hors des accords, modulations, chiffrage, etc. Ce sont vraiment des textes qu'on peut placer n'importe où. À commencer pour définir les parties de l'ouvrage (« Introduction », « Coda », etc.).

Dans un texte, il est impérative de remplacer toutes les espaces par des traits plats (on les obtient, sur mac, à l'aide de Maj+tiret).

Par exemple, pour écrire sur la partition :

**Premier couplet**

Il faut définir la ligne :

**ligne Premier\_couplet type=partie y= 50 x=200**

Note : ici, c'est le type **partie** qui fera que le texte s'écrit de travers, dans une boîte.

## Les types de textes

- Les parties
- Les modulations
- Les mesures

type	anglais	Description
partie	part	Titre de partie, comme Exposition ou Coda
mesure	measure	Numéro de mesure, dans un carré.
modulation	(id.)	Marque de modulation, en haut de partition, inclinée.

## Les parties

Les marques de partie s'indiquent avec le tag **partie** (ou **par** ou **part**). Ce sont des textes dans des boites inclinées qui ont cet aspect :



Figure 4: Marque de partie

## Les mesures

Les numéros de mesure, s'il ne sont pas indiqués sur la partition elle-même, peuvent être ajoutés à l'aide du tag **measure** (ou **measure**, ou **mes**), suivant du numéro de mesure puis des coordonnées.

## Les modulations

On peut mettre un texte au-dessus de la barre inclinée (en général la tonalité vers laquelle on module) et un texte en dessous (en général la fonction de cette tonalité).

Pour séparer les deux textes, on utilise tout simplement la barre inclinée, appelée « balance ». Ainsi, pour obtenir :

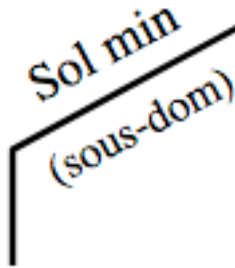


Figure 5: Modulation avec sous-titre

... on utilisera simplement :

```
modulation Sol_min/(sous-dom.) x=200 y=300
```

ou

```
mod Sol_min/(sous-dom.) 200 300
```

### Dessiner des lignes

Les lignes se définissent par `line` ou `ligne`.

Le premier élément définit le **type** de la ligne. On trouve les types suivants.

U ou  __	Ligne inférieure et trait vertical avant/après
N ou  ---	Ligne supérieure et trait vertical avant/après
L ou  __	Ligne inférieure et trait vertical avant
K ou  ---	Ligne supérieure et trait vertical avant
V ou __	(Virgule) Trait inférieur et trait vertical après
^ ou ---	(Virgule inversée) Trait supérieur et trait vertical après

On peut ensuite définir sa taille et sa position avec les lettres habituelles `x` (position horizontale), `y` (position verticale) et `w` (largeur en pixels).

### Procédure de découpage de la partition

Voyons quelques méthodes de découpage de la partition en « images-systèmes ». Je les présente ici de la meilleure à la moins bonne. Cette qualité a été définie fonction des deux critères suivants :

- rapidité d'exécution,
- précision du découpage.

### Avec capture sélection dans Aperçu (Mac)

Méthode la plus rapide, mais également la moins précise. Ce manque de précision oblige parfois à reprendre des systèmes pour mieux les découper. Cependant, elle est tellement plus rapide que les autres que je la privilégie sans problème.

- Ouvrir la partition PDF dans l'application Aperçu,
- jouer **CMD Maj 4** pour activer la sélection par souris,
- sélectionner la zone de la partition à capturer (un système),
- recommencer l'opération pour tous les systèmes,
- récupérer les captures sur le bureau — sauf si l'astuce ci-dessous (1) a été utilisée — et les mettre dans le dossier **images** de votre analyse,
- modifier les noms des fichiers — sauf si vous avez utilisé l'astuce ci-dessous (1) — en les indiquant de 1 (ou 0) à N pour les insérer plus facilement dans l'analyse.

(1) Astuce : pour aller encore plus vite, vous pouvez :

- utiliser l'utilitaire Muscat **change\_folder\_captures** pour définir le dossier des captures écran ou consulter la procédure décrite ici. Vos captures iront directement dans ce dossier,
- effectuer les captures,
- utiliser l'utilitaire Muscat **rename\_images** pour renommer instantanément vos fichiers.

Note : vous pouvez voir ou revoir la procédure dans les tutoriels consacrés sur ma chaîne YouTube.

### Avec sélection rectangulaire dans Aperçu (Mac)

Une méthode qui ressemble à la précédente et permet d'être plus précis. Mais cette précision se fait au détriment du temps, notamment pour l'enregistrement des fichiers images.

- ouvrir la partition PDF dans Aperçu,
- choisir la sélection rectangle (p.e. Outils > Sélection rectangulaire),
- sélectionner le système grossièrement,
- ajuster parfaitement la sélection à l'aide des poignées,
- copier la sélection (**CMD C**),
- activer la combinaison **CMD N** pour créer une nouvelle image à partir du presse-papier,
- enregistrer l'image (**CMD S**) avec le nom voulu, dans le dossier voulu, en choisissant le format voulu.



### Avec Aperçu, sélection souris et rectangle (Mac)

On peut bien entendu imaginer une méthode intermédiaire qui reprendrait les deux méthodes précédentes. Lorsque la découpe est facile, on utilise la première, lorsque la découpe demande plus de précision, on privilégie la seconde.

### Avec MuScaT et convert

C'est une méthode qui souffre parfois d'un manque de qualité de rendu.

On tire déjà les images du PDF à l'aide de la commande à jouer dans le Terminal (adapter la qualité du traitement en fonction du résultat) :

```
# Se trouver dans le dossier contenant la partition (cd ...)
convert[ options] partition.pdf partition.jpg # ou .png
```

Autant d'images que de pages sont produites.

On insert la première dans le code du fichier `tags.js`, avec l'option `crop image` :

```
# Dans tags.js
option('crop image')
Tags=`
partition partition-0.jpg
`;
```

On ouvre le fichier `partition.html` dans Firefox.

Maintenant, il suffit de sélectionner, à la souris, la zone de l'image à prendre puis de coller le code du presse-papier dans la console du Terminal. Puis de jouer ce code.

Répéter l'opération avec chaque système, puis avec chaque page de la partition.

### Avec Gimp/Photoshop (ou autre logiciel de traitement de l'image)

Si Gimp présente une précision de découpage inégalable, l'application offre en revanche la méthode la plus chronophage, même avec l'habitude du logiciel.

- ouvrir le PDF dans Gimp,
- sélectionner chaque système en le découpant,
- le placer en haut,
- « cropper » l'image à la taille du plus haut système,
- exporter chaque image-système (avec le bon nom).

Ce mode d'emploi n'étant pas destiné à maîtriser Gimp, je vous renvoie au manuel d'utilisation de l'application.

---

## Options

- Option « code à côté »
- Option « découpe image »
- Option « lignes de repère »
- Option « espacement images »
- Option « marge haut »
- Option « marge gauche »

Comme les tags et les partitions, les options se règle dans le fichier `tags.js`. On utilise tout naturellement la fonction `option` (ou `options`) avec en argument les options à activer.

Ci-dessous, par exemple, on active l'option `guide` qui affiche deux lignes repère déplaçables pour aligner des éléments à la souris (ou par magnétisation).

```
// Dans tags.js
option('guide');
Tags=`
  ...
`;
```

Dans la méthode `option`, on peut passer toutes les options les unes à la place des autres, ou utiliser plusieurs fois la méthode `option`. Les deux formulations suivantes sont équivalentes :

```
// Dans tags.js
option('guide', 'code', 'marge haut', 100);
```

... équivaut à :

```
// Dans tags.js
option('guide');option('code');option('marge haut', 100);
```

Note : les points virgules sont optionnels.

Vous noterez qu'il existe deux types d'options. Les options dites « booléenne » qu'on active simplement en indiquant leur nom en argument (par exemple `guide` ou `code`) et il y a les options non booléennes qui attendent une valeur précise (par exemple `marge haut` attend la valeur de cette marge haut).

Dans les arguments de la méthode `option`, la valeur des options non booléennes doit suivre immédiatement le nombre de l'option :

```
// Dans tags.js
```

```
option('marge haut', 100);
```

### Option « code à côté »

Option : code beside, code à côté

Type : booléen

L'option « code à côté » permet d'avoir le fichier contenant le code juste à côté de la partition, ce qui est très pratique pour le modifier sans avoir à changer d'application. On le voit ci-dessous dans la boîte noire.



Figure 6: Code à côté de la partition

### Option « découpe image »

Option : crop image, découpe image

Type : booléen

Cette option fait passer dans un mode d'utilisation qui va permettre de découper l'image de façon aisée (par simple [copié]-collé).

### Option « lignes de repère »

Option : repères, repere, lines of reference

Type : booléen

Ajoute une ligne horizontale et une ligne verticale qu'on peut déplacer et qui peuvent servir de guide, de repère, pour placer les TAGs.

### Option « Espacement entre images »

Option : `espacement images, space between scores`

Type : non booléen, la valeur est le nombre de pixels

Permet de régler l'espacement en pixels entre deux images lorsque l'écriture séquentielle des images a été adoptée.

```
// Dans tags.js
option('espacement images', 100);
Tags=`
  ...
`;
```

Avec le code ci-dessus, l'espace entre les différents systèmes sera de 100 pixels.

### Option « marge haut »

Option : `marge haut, top first score`

Type : non booléen, la valeur est le nombre de pixels

Lors de l'écriture séquentielle des images, cette valeur permet de déterminer à quelle hauteur doit être placée la première image (le premier système ou la partition).

```
// Dans tags.js
option('marge haut', 200);
Tags=`
  ...
`;
```

Avec le code ci-dessus, la première image de partition sera placée à 200 pixels du haut.

### Option « marge gauche »

Option : `marge gauche, left margin`

Type : non booléen, la valeur est le nombre de pixels

Lors de l'écriture séquentielle des images, cette valeur détermine la marge gauche où placer l'image (son `x`).

```
// Dans tags.js
option('marge gauche', 50);
Tags=`
```

```
...  
;
```

Avec le code ci-dessus, toutes les images de la séquence seront placées à 50 pixels de la gauche.

---

## Utilitaires

L'application **MuScaT**, comme tout bon vin, est fournie avec quelques utilitaires pour se faciliter la vie, en tout cas sur Mac. En voici la liste avec leur mode d'utilisation.

### Renommage des fichiers images (Mac/Unix)

Ce script, qui se trouve dans le dossier `utils` de l'application, permet de renommer les images d'un dossier de façon cohérente et indexée.

Pour utiliser ce script :

- ouvrir l'application Terminal,
- rejoindre (commande `cd`) le dossier de l'application MuScaT (ATTENTION : ça n'est pas le dossier de l'analyse, ici, c'est bien le dossier de l'application),
- se placer dans le dossier utilitaires (`cd utils`)
- taper `./rename_images.rb -h` et la touche Entrée pour tout savoir du script.

### Changement du dossier des captures écran (Mac)

Par défaut, les captures d'écran sont enregistrés sur le bureau. Ça n'est pas gênant en soit, il suffit de les glisser ensuite dans le dossier `image` de l'analyse. Mais si on veut encore gagner du temps, ce script permet de changer le dossier de destination.

Voici la procédure :

- ouvrir l'application Terminal,
- rejoindre (commande `cd`) le dossier `utils` de l'application MuScaT (ATTENTION : ça n'est pas le dossier de l'analyse, ici, c'est bien le dossier de l'application),
- taper `./change_folder_captures.rb -h` et la touche Entrée pour tout savoir du script.

Pour remettre la valeur par défaut (le bureau), jouer simplement `./utils/change_folder_captures.rb` sans aucun autre argument.

### Création d'une nouvelle analyse (Mac)

Le script `create.rb` permet de créer une nouvelle analyse dans le dossier `_analyses_` de **MuScaT**.

- ouvrir l'application Terminal,
- rejoindre (commande `cd`) le dossier `utils` de l'application MuScaT,
- puis, au choix :
  - taper `./create.rb -h` et la touche Entrée pour tout savoir du script,
  - taper `./create.rb "Ma nouvelle analyses" -o` pour créer l'analyse et l'ouvrir dans le finder.

Notez que pour l'activer, il faut l'ouvrir dans le navigateur avec le script `./analyse.rb`.

### Activation d'une analyse (Mac)

Le script `analyse.rb` permet d'activer une analyse se trouvant dans le dossier `_analyses_` de **MuScaT**.

- ouvrir l'application Terminal,
- rejoindre (commande `cd`) le dossier `utils` de l'application MuScaT,
- puis, au choix :
  - taper `./analyse.rb -h` et la touche Entrée pour tout savoir du script.
  - taper `./analyse.rb` pour obtenir la liste des analyses et en choisir une,
  - taper `./analyse.rb "Mon_analyse"` pour ouvrir l'analyse qui commence par ce titre.