

---

# **Développement d'outils ou matériel d'enseignement de l'informatique**

## **Développement d'un outil informatique démontrant visuellement la perception d'une liste pour un ordinateur**

*Collège du sud, Travail de maturité Version intermédiaire*

**Grégoire Geinoz**

déc. 09, 2021



---

## Table des matières

---

<b>1</b>	<b>Introduction (version Markdown / MyST)</b>	<b>1</b>
1.1	Informations à inclure pour un TM de type « outil » . . . . .	1
1.2	Informations à inclure pour un TM de type « tutoriel » . . . . .	1
<b>2</b>	<b>Question</b>	<b>3</b>
2.1	Formulation du problème . . . . .	3
2.2	Solutions envisageables . . . . .	3
<b>3</b>	<b>Titre du chapitre 1</b>	<b>5</b>
3.1	Titre 1 . . . . .	5
<b>4</b>	<b>Titre du chapitre 2</b>	<b>7</b>
4.1	Titre 1 . . . . .	7
<b>5</b>	<b>Tutoriel Sphinx / MyST</b>	<b>9</b>
5.1	Fonctionnement de la toolchain Sphinx . . . . .	9
5.2	Les bases de la rédaction avec Sphinx . . . . .	11
5.3	Insérer du code . . . . .	12
5.4	Fonctionnalités avancées . . . . .	13
<b>6</b>	<b>Indices and tables</b>	<b>15</b>



---

## Introduction (version Markdown / MyST)

---

---

### Information

Ce chapitre d'introduction doit être complété avec les informations concernant le chapitre d'introduction au TM.

---

À compléter ... avec les informations concernant le chapitre d'introduction au TM.

## 1.1 Informations à inclure pour un TM de type « outil »

Si le TM est du type « outil d'enseignement », ce chapitre doit contenir les informations suivantes :

- Présentation rapide de l'outil
- Intérêt de l'outil (ce qu'il apporte) et aperçu des solutions similaires existantes s'il y en a
- Aperçu des technologies utilisées
- Configuration matérielle requise pour compiler / utiliser l'outil
- Connaissances requises pour utiliser l'outil
- Connaissances requises pour comprendre le fonctionnement de l'outil

## 1.2 Informations à inclure pour un TM de type « tutoriel »

Si le TM est du type « tutoriel », ce chapitre doit contenir les informations suivantes :

- Présentation très rapide du sujet / domaine
- Si le tutoriel montre comment développer un projet, présenter très rapidement le projet
- Objectifs pédagogiques du tutoriel
- Aperçu des technologies utilisées
- Configuration matérielle requise pour compiler / utiliser le projet
- Connaissances requises pour comprendre le projet

Ce chapitre doit présenter rapidement le projet, son intérêt, les technologies utilisées.



## 2.1 Formulation du problème

Le problème réside dans la gestion des déplacements des cartes en JavaScript. En effet, une animation de carte dure un certain temps et pendant ce laps de temps, le code doit être « bloqué ». C'est à dire que si une autre animation devait s'exécuter, elle devrait d'abord attendre que l'animation précédente soit terminée.

Or l'animation est gérée avec Phaser dans une fonction « `update()` » asynchrone, ce qui implique que la suite du code est exécuté en parallèle de l'animation, rien n'empêche donc de lancer plusieurs animations simultanément. De plus, le code principal n'est pas conscient qu'une animation est en cours, il ne peut donc pas attendre que l'animation soit finit avant d'exécuter la suite du programme.

## 2.2 Solutions envisageables

En écrivant précisément le problème, les solutions envisageables viennent plus facilement à l'esprit. Il se trouve que l'une d'entre elle m'est venu à l'esprit en écrivant.

En effet, sans se prendre la tête en utilisant les nouvelles normes « `async/await` » apparues de la version ES7 de JavaScript, l'utilisation d'une « Promise » à l'air envisageable afin d'attendre une « promesse » retournée par l'animation avant d'exécuter la suite du code.

Une autre approche est à prendre en compte, il s'agit de stocker les animations dans une liste. Ainsi, il serait donc possible de jouer les animations les une après les autres, selon l'ordre dans lequel elles ont été ajoutés à la liste.

Enfin, un mélange de ces deux solutions semblerait permettre de résoudre le problème aisément.





# CHAPITRE 3

---

## Titre du chapitre 1

---

### 3.1 Titre 1

{ref}``

```
1 const variable = "Var1";
```

#### 3.1.1 Titre 2



## CHAPITRE 4

---

Titre du chapitre 2

---

### 4.1 Titre 1

#### 4.1.1 Titre 2



---

### En bref

Sphinx est le système de documentation du langage Python (<https://www.sphinx-doc.org/en/master/>).

Ce tutoriel, livré avec le kit de démarrage Sphinx pour la rédaction du travail écrit de votre TM, vous permet d'acquérir rapidement les bases de la syntaxe Sphinx.

---

## 5.1 Fonctionnement de la toolchain Sphinx

Avant de commencer, il faut comprendre le fonctionnement général de Sphinx. Il s'agit d'une « toolchain », à savoir un ensemble d'outils qui permettent de transformer des fichiers Markdown en documents formatés (soit HTML pour le Web ou LaTeX pour produire un PDF).

Pour les personnes qui connaissent un peu le LaTeX, Sphinx est un système un peu analogue : on écrit du code et un outil s'occupe de tout mettre en page et faire la reliure de manière professionnelle. D'ailleurs, Sphinx permet de générer du LaTeX pour générer un document PDF.

Au contraire de Microsoft Word, où l'on voit directement le résultat final lorsqu'on écrit le contenu, Sphinx utilise un paradigme semblable à celui utilisé en LaTeX : on écrit du code avec une syntaxe particulière qui est ensuite transformé dans le résultat final par la toolchain.

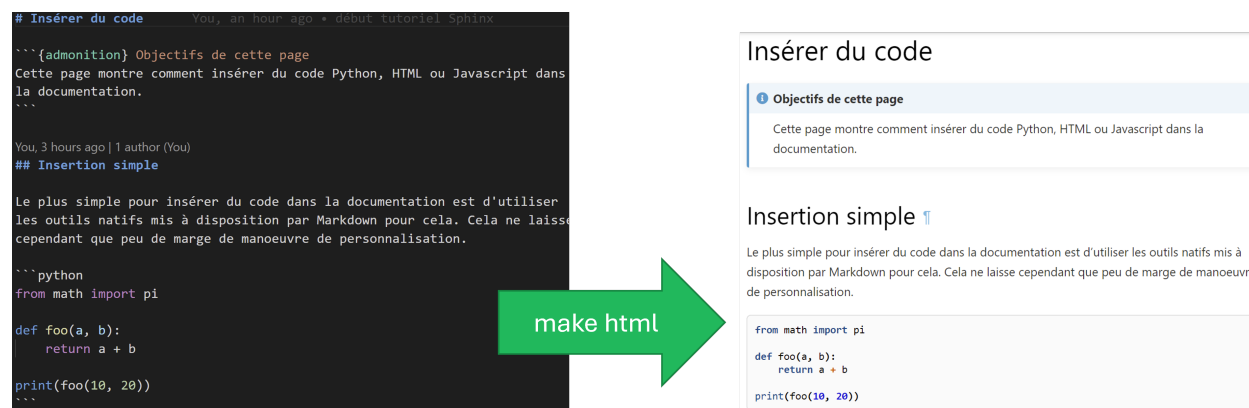


Fig. 1 – Processus de génération du HTML avec Sphinx

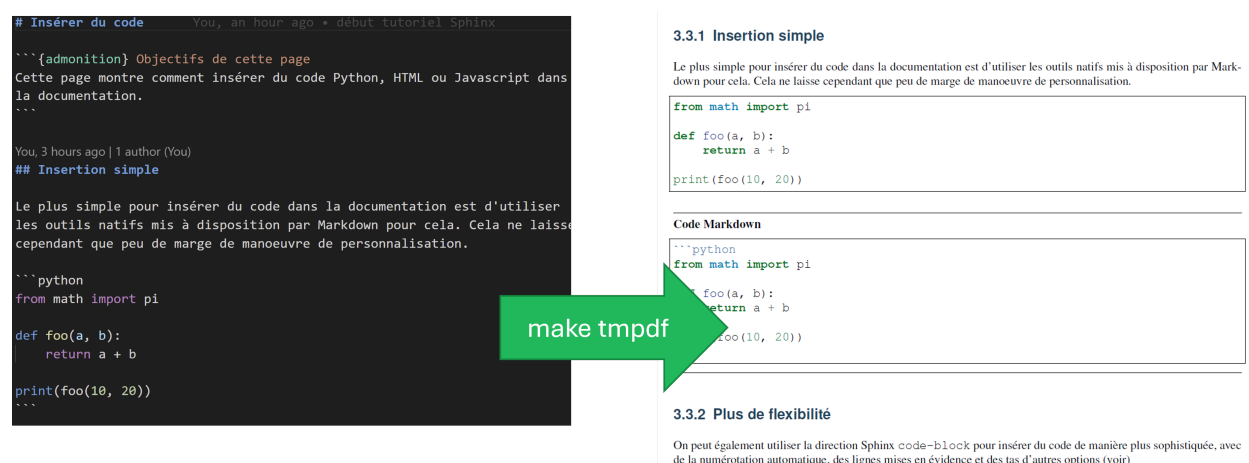


Fig. 2 – Processus de génération du PDF avec Sphinx

### 5.1.1 Les avantages de Sphinx par rapport à Word

Par rapport à Word, Sphinx présente les avantages suivants

- Sphinx est un outil Open-Source, ce qui signifie qu'on peut l'adapter sans problème à n'importe quel besoin. De plus, il n'y a aucun risque que le système ne soit tout-à-coup plus supporté, car une importante communauté s'occupe de le faire évoluer et de corriger les bugs détectés.
- On écrit de la documentation dans des fichiers qui ne contiennent que du texte, comme une sorte de « code source ».
- On écrit le contenu dans Visual Studio Code ou un autre éditeur. On peut donc utiliser toute la puissance de tels éditeurs pour optimiser la rédaction du contenu.
- Le contenu est très facilement versionnalisable (utilisation de git et Github).
- Permet d'intégrer très facilement les éléments suivants qui sont très courants dans un travail de maturité en informatique
  - Des mathématiques
  - Du code source
  - Des éléments interactifs (quiz, programmes qui s'exécutent directement dans le navigateur, etc. )
- À partir d'une seule source, on peut générer des documents pour différentes cibles (Site Web, Document PDF, e-book pour une liseuse, ...)
- Le système est extensible et facilement personnalisable
- Lorsqu'on écrit un document pour Sphinx, il n'est pas nécessaire de se préoccuper de la mise en page et de la mise en forme du contenu. On se contente d'écrire

### 5.1.2 Les désavantages de Sphinx par rapport à Word

- L'utilisation de Sphinx est moins intuitive que celle de Word au début.
- Word est un traitement de texte de type « WYSIWYG » (What You See Is What You Get). Lorsqu'on écrit le contenu dans Word, on voit directement le résultat final. Sphinx est un système de rédaction de type « WY-SIWYM » (What You See Is What You Mean). Comme on ne voit pas directement le résultat final, il faut être attentif lors de la relecture du résultat final. On peut parfois avoir de sacrées surprises, en raison d'une erreur de syntaxe par exemple.
- La correction orthographique est moins facile qu'en Word, car il n'existe pas d'outil parfaitement intégré qui permette de faire la correction de manière automatique. Il est toutefois possible de passer du Sphinx à Word (avec un rendu dégradé), pour utiliser le correcteur automatique de Microsoft Word.

## 5.2 Les bases de la rédaction avec Sphinx

---

### En cours de rédaction

Cette section est en cours de rédaction et sera complétée à l'avenir.

---

## 5.2.1 Fichiers Markdown ou RestructuredText

Le contenu à proprement parler est écrit dans des fichiers textes au format Markdown / MyST (voir)

---

**Astuce :** Pour écrire le TM, il est conseillé d'utiliser le format MyST qui

dispose d'une extension Visual Studio Code très pratique (<https://marketplace.visualstudio.com/items?itemName=ExecutableBookProject.myst-highlight>).

---

## 5.2.2 Bien démarrer

Pour le moment, il est recommandé de travailler sur Gitpod pour rédiger votre TM, à moins que vous ayez Linux installé sur votre machine et que vous soyez à l'aise avec.

---

### Information

Nous allons apprendre à utiliser Linux ces prochains mois au cours d'OC.

---

En attendant que ce tutoriel soit plus fourni, reportez-vous à la documentation officielle de MyST

- <https://myst-parser.readthedocs.io/en/latest/sphinx/intro.html#intro-writing>
- <https://myst-parser.readthedocs.io/en/latest/syntax/syntax.html#>
- <https://myst-parser.readthedocs.io/en/latest/syntax/optional.html>
- <https://myst-parser.readthedocs.io/en/latest/syntax/reference.html>

Il peut également être utile de consulter la documentation de Sphinx pour savoir comment utiliser les fonctionnalités avancées.

- <https://www.sphinx-doc.org/en/master/index.html>

## 5.3 Insérer du code

---

### Objectifs de cette page

Cette page montre comment insérer du code Python, HTML ou Javascript dans la documentation.

---

### 5.3.1 Insertion simple

Le plus simple pour insérer du code dans la documentation est d'utiliser les outils natifs mis à disposition par Markdown pour cela. Cela ne laisse cependant que peu de marge de manoeuvre de personnalisation.

```
from math import pi

def foo(a, b):
    return a + b

print(foo(10, 20))
```

---

### Code Markdown



```
```python
from math import pi

def foo(a, b):
    return a + b

print(foo(10, 20))
```
```

### 5.3.2 Plus de flexibilité

On peut également utiliser la direction Sphinx `code-block` pour insérer du code de manière plus sophistiquée, avec de la numérotation automatique, des lignes mises en évidence et des tas d'autres options (voir)

```
1 from math import pi
2
3 def foo(a, b):
4     return a + b
```

#### Code Markdown

```
```{code-block} python
---
emphasize-lines: 1
linenos: true
---
from math import pi

def foo(a, b):
    return a + b
```
```

### 5.3.3 Insérer du code depuis un fichier externe

Il est également possible d'inclure du code dans la documentation depuis un fichier externe, au lieu d'avoir à écrire copier le code directement dans le fichier `.md`.

Titre 2

Titre 3

## 5.4 Fonctionnalités avancées

### 5.4.1 Encadrés

#### Encadré mis en évidence

Ceci est un encadré qui sort du fil du texte (on dit que cet encadré est « flottant », car il flotte sur le texte principal).

La directive Sphinx `sidebar` permet d'insérer certaines informations dans un encadré (dans le PDF) et dans une sorte de panneau flottant dans la version HTML.

$$f(x) = 2x^2 + 1$$

Il faut commencer par placer le `sidebar` dans le code Markdown et ensuite placer le contenu qui viendra se placer à gauche de la page.

Paragraphe est écrit de manière normale, au fil du texte.

- On pourrait remplir avec du contenu .... mais ce contenu n'est pas très intéressant. Ce contenu peut faire plusieurs lignes dans le fichier Markdown. C'est tout de même un peu long.

**Avertissement :** Il ne faut pas trop abuser des « sidebars », car le placement n'est pas du tout le même sur la version HTML en ligne et dans la version PDF.

Il vaut éviter également de mettre trop de contenu dans les `sidebar`.

### 5.4.2 Direction `raw`

---

**Astuce :** La directive `raw` permet de différencier le contenu présenté dans la version HTML et celui présent dans la version PDF. Elle permet également d'insérer du contenu tel quel pour profiter de toutes les fonctionnalités du format cible (par exemple le HTML ou le LaTeX).

---

Parfois, il est utile d'inclure un certain contenu (images, vidéos, ...) dans la version HTML en ligne, mais de ne pas inclure ce contenu dans la version PDF. C'est typiquement le cas pour des vidéos ou d'autres éléments interactifs.

En revanche, ce contenu n'apparaîtra que dans le LaTeX et pas dans le HTML.

---

**Astuce :** L'avantage de la directive `raw` est de pouvoir insérer n'importe quel code HTML, aussi exotique soit-il.

---

## CHAPITRE 6

---

### Indices and tables

---

- `genindex`
- `search`