

# Beste de savoir

Mon CV en TOML

---

4 novembre 2022



# Table des matières

	Introduction . . . . .	1
1.	Écrire un <code>cv.toml</code> . . . . .	1
2.	Conclusion . . . . .	4

## Introduction

J'ai décidé de rédiger mon nouveau CV en [TOML](#) .

Il s'agissait au début d'une blague, mais après réflexion j'ai voulu tester et rédiger sérieusement mon nouveau CV en TOML, en essayant d'utiliser au mieux ce que le langage propose sémantiquement et syntaxiquement. Le langage TOML se prévaut d'être un langage aussi bien facile à écrire qu'à lire pour l'humain, en plus de l'être pour la machine. Naturellement, il faut donc que ce CV soit distribuable et lisible par les humains tel quel, c'est à dire en TOML brut.

## 1. Écrire un `cv.toml`

Comme n'importe quel document, ce CV a un titre. Et comme n'importe quel CV, on va commencer par se présenter avec diverses métadonnées jugées pertinentes selon le profil.

```


```

Poursuivons avec l'expérience professionnelle. Voici la structure que j'ai choisie, elle reprend les attributs classiques que l'on retrouve dans un CV. Pour les dates de début et de fin, `start` et `end`, on utilise le type natif de [date](#) de TOML.

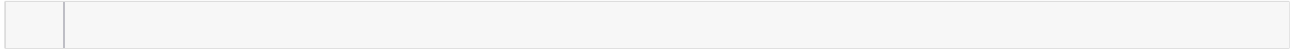
```


```

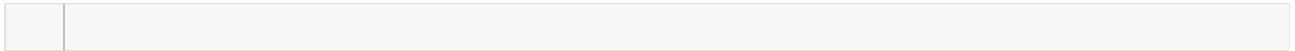
J'ai bien de la chance, peu de mes expériences sont dignes qu'on leur accorde plus de quelques mots dans le champ `desc` (description), faisant ainsi tenir leur description sur une ligne sous 80–100 caractères. Mais autrement, TOML aurait su gérer la situation avec des chaînes de caractères sur plusieurs lignes, ainsi que les tableaux sur plusieurs lignes, ce qui facilite la lisibilité humaine.

Exemple (fictif) d'une de mes expériences dans une autre vie:

## 1. Écrire un cv.toml

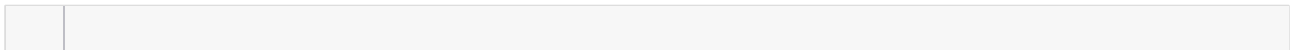


Cela montre le principe général. Pour chaque type d'entrée, on a des champs à renseigner et le CV consiste surtout en une table de listes énumérant d'autres tables d'un même «type». Voici des exemples de ce que l'on pourrait avoir comme autres types d'objets, ils parleront sûrement d'eux-mêmes.



C'est assez confortable de pouvoir imbriquer «à plat» certaines catégories, ici `skills.tech` et `skills.other`.

Le document final paraît assez «plat», ce qui le rend agréable à lire humainement, mais en contrepartie cela laisse peu ressortir la structure sous-jacente. Le voici dans son intégralité, suivi du JSON équivalent pour mieux se faire une idée de la structure.



Équivalent en JSON:

```
1 {
2   "title": "Sgble's Curriculum Vitae ",
3   "lang": "en",
4   "me": {
5     "name": "Sgble",
6     "location": "Paris, FR",
7     "about": "I don't know what to say, I'm shy!",
8     "email": "sgble@example.com",
9     "github": "@sgble-example"
10  },
11  "jobs": [
12    {
13      "title": "Benevolent Chairman for Life",
14      "company": "Sgble Corp",
15      "type": "self-employed",
16      "start": "2012-02-01",
17      "location": "Paris, FR",
18      "desc": "
19        I founded Sgble Corp to solve great challenges for the next decades
20      ",
21      "tech": [
22        "Office 360",
23        "Rolodex",
24        "MyOpenConciergerie"
25      ]
26    }
27  ],
28 }
```

## 1. Écrire un cv.toml

```
25     {
26       "title": "Software Analyst & Architect (not Developer)",
27       "company": "Yet Another Tech Company",
28       "type": "full-time",
29       "start": "2010-07-03",
30       "end": "2012-02-01",
31       "location": "Paris, FR",
32       "desc": "Helped write code for the flagship product.",
33       "tech": [
34         "Python",
35         "Go",
36         "Kubernetes",
37         "Redis",
38         "PostgreSQL"
39       ]
40     }
41 ],
42 "skills": {
43   "tech": [
44     {
45       "skill": "Python",
46       "level": "Advanced"
47     },
48     {
49       "skill": "Go",
50       "level": "Intermediate"
51     }
52   ],
53   "other": [
54     {
55       "skill": "Accounting",
56       "level": "Intermediate"
57     },
58     {
59       "skill": "Project Management",
60       "level": "Advanced"
61     }
62   ]
63 },
64 "spoken-languages": [
65   {
66     "lang": "fr_FR",
67     "level": "Native"
68   },
69   {
70     "lang": "en_US",
71     "level": "Somewhat okay"
72   },
73   {
74     "lang": "fr_BE",
```

## 2. Conclusion

```
75     "level": "Elementary"  
76   }  
77 ]  
78 }
```

## 2. Conclusion

Personnellement j'aime bien mon CV en TOML. Mais il faut quand même admettre qu'un bon thème de coloration syntaxique ne sera pas un luxe pour en faciliter la lecture. Reconnaissons aussi que les commentaires—que TOML a le bon goût d'autoriser—contribuent aussi à la lisibilité en séparant les sections.

Le rédiger ne m'a posé aucun problème. Certes, la structure est triviale, mais en YAML—qui aurait pu peut-être être aussi lisible—j'aurais été capable de mettre la mauvaise indentation quelque part, cassant ainsi la cohérence du document. Quant à JSON ou XML, je n'aurais pas fait un document valide du premier coup.

Pour ce qui est de la lisibilité, je vous laisse en qualité de juges. Cependant, je vous invite à utiliser votre éditeur de texte favori pour juger, car le thème TOML de ZdS n'est peut-être pas celui qui le rendra le mieux.

*i*

Le saviez-vous? La nouvelle version de Python 3.11 inclut `tomllib` dans sa bibliothèque standard, vous permettant ainsi de traiter des CV structurés en TOML automatiquement. Plus d'informations dans l'article d'entwanne. ↗