

法情報教育の最前線

日本の進むべき将来像を模索する



明治学院大学名誉教授
加賀山 茂

Table of Contents

■ LegalTech

- [LegalTech in 5 forces](#)
- [A simple example of XML](#)

■ XML & Python

- [Why XML in LegalTech?](#)
 - [Application Neutrality](#)
 - [Hierarchical Structure](#)
 - [Platform Neutrality](#)
 - [International Language Support](#)
- Why Python with XML?
 - [Easy to read and maintain](#)
 - [Exploratory programming in an interactive interpreter](#)
 - [Portability without restrictions](#)
 - [Powerful but accessible object-orientation](#)
 - [Demonstration with Python](#)

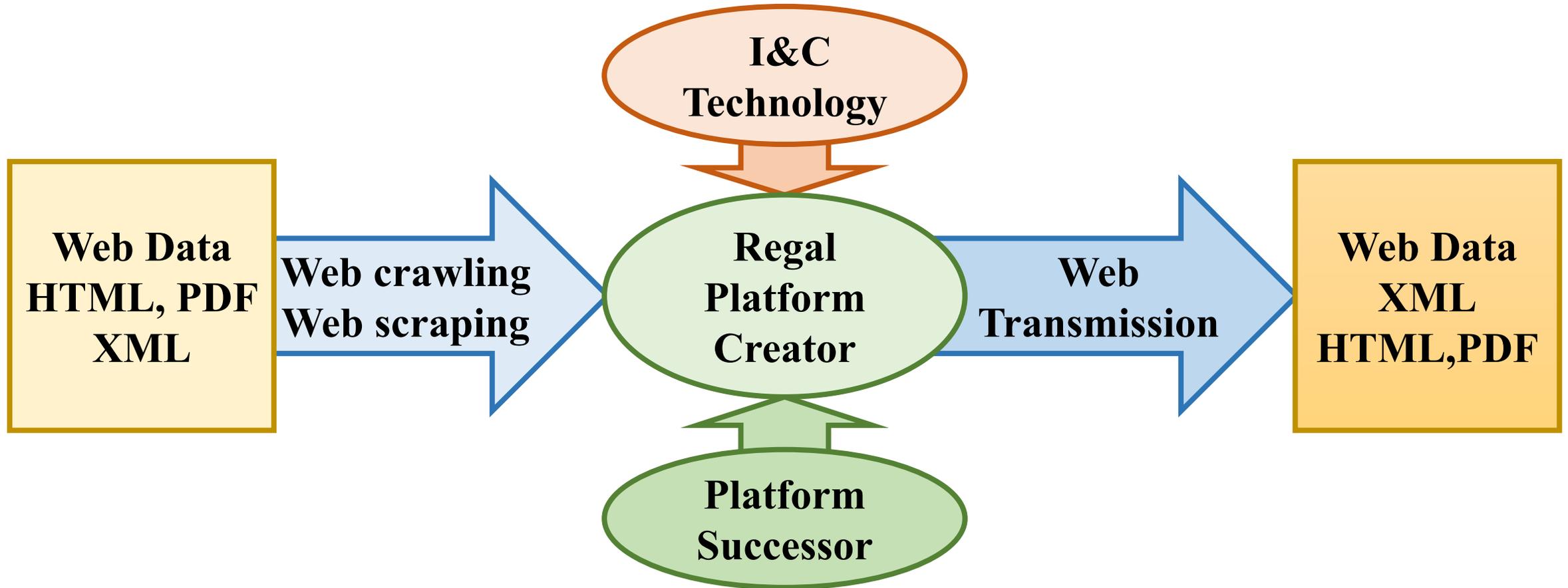
■ Data Science

- [DataTech\(佐々木・志田『データテック』\(2019\)の紹介と検討\)](#)
 - [序章 なぜ『動かないAI』が続出するのか](#)
 - [第1章 第4次産業革命とデジタルトランスフォーメーション](#)
 - [第2章 XMLルネサンス\(再興\)の必然](#)
 - [第3章 新たな世界を開くXMLの自動変換技術](#)
 - [第4章 XMLの自動変換技術, ポイント解説ー 第4次産業革命を導くデータの在り方](#)
 - [第5章 XMLデータ活用で業界は一変する](#)
 - [第6章 データ志向で描くデジタル国家像](#)
 - [第7章 「データの解放」へ時代が動く](#)
- [LegalTech\(佐々木・リーガルテック\(2017\)の紹介\)](#)

■ [Bibliography](#)



LegalTech in 5 forces



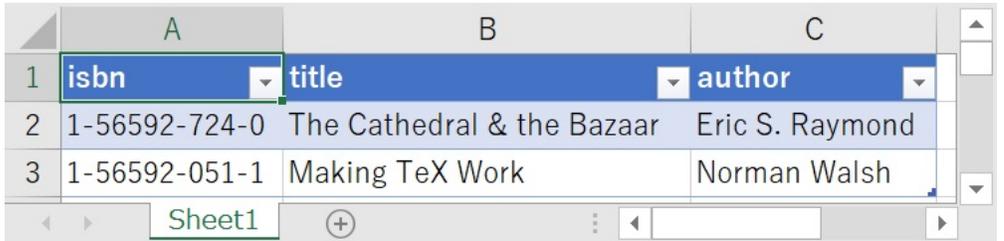
A simple example of XML

XML data

```
<?xml version="1.0"?>
- <catalog>
  - <book isbn="1-56592-724-0">
    <title>The Cathedral & the Bazaar</title>
    <author>Eric S. Raymond</author>
  </book>
  - <book isbn="1-56592-051-1">
    <title>Making TeX Work</title>
    <author>Norman Walsh</author>
  </book>
  <!-- imagine more entries here... -->
</catalog>
```

Excel data

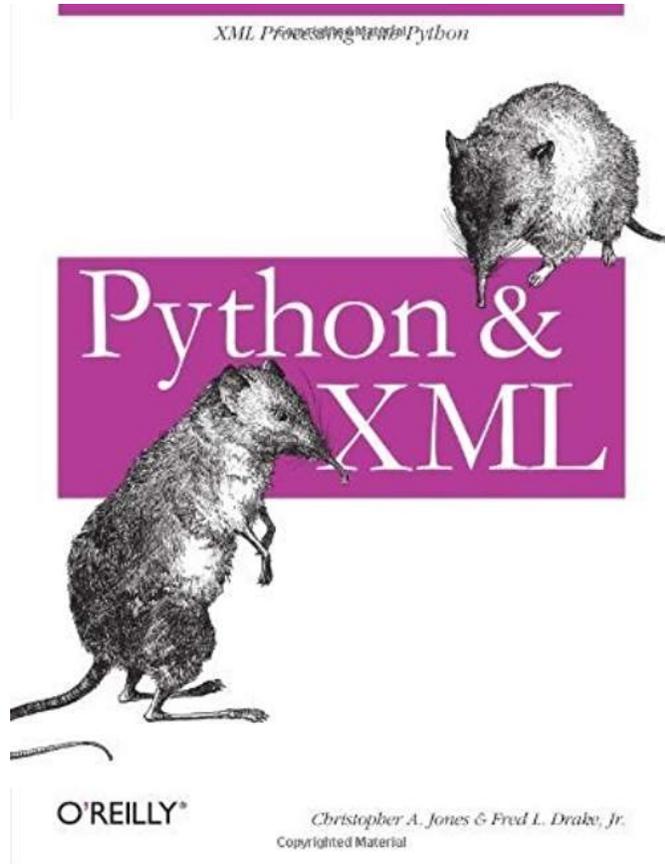
Microsoft Excel can open any XML file with format of itself like as follows:



	A	B	C
1	isbn	title	author
2	1-56592-724-0	The Cathedral & the Bazaar	Eric S. Raymond
3	1-56592-051-1	Making TeX Work	Norman Walsh



Why XML & Python's combination is suitable for LegalTech?



■ Christopher A. Jones = Fred L. Drake, Jr., "Python & XML", O'Reilly(2002)

■ Table of contents

1. Python and XML
2. XML Fundamentals
3. The Simple API for XML
4. The document Object Model(DOM)
5. Querying XML with Xpath
6. Transforming XML with XSLT
7. XML Validation and Dialects
8. Python Internet APIs
9. Python, Web Services, and SOAP
10. Python and Distributed System Design



Why XML? (1/4)

■ 1. Application Neutrality

- XML is both human- and machine-readable.
 - Microsoft Word document , for example, cannot be read with a text editor.
 - The Word document is in a binary (computer readable only) format.
 - XML is a text file document, everyone can read it without expensive application software.



Why XML? (2/4)

■ 2. Hierarchical Structure

- XML is hierarchal, like a legal sentences, and allows you to choose your own tag names.

■ e.g.

- `<?xml version="1.0"?>`

- `<catalog>`

- `<book isbn="1-56592-724-0">`

- `<title>The Cathedral & the Bazaar</title>`

- `<author>Eric S. Raymond</author>`

- `</book>`

- `</catalog>`

- The script of left side is quit different from HTML.

- In XML, you are free to create elements of any type, and stack other elements within those elements

- You can interpret these script like this:

- book

- book.isbn = 1-56592-724-0

- book.title = The Cathedral & the Bazaar

- book.author = Eric S. Raymond



Why XML? (3/4)

■ 3. Platform Neutrality

- XML is cross-platform.
- XML can be readily moved from one type of machine to another without loss of information.
- A chapter of a technical book can be written by a programmer on his/her favorite flavor of Unix, and then sent to a publisher using book composition software on a Macintosh.



Why XML? (4/4)

■ 4. International Language Support

- Standard bodies all over the world have come up with ways that computers can interchange text written in their national language at a time.
- Since XML documents are Unicode documents, the language of the world are supported.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <カタログ>
  - <書籍 ISBN="1-56592-724-0">
    <タイトル>伽藍とバザール</タイトル>
    <著者>エリック S. レイモンド</著者>
  </書籍>
  - <書籍 ISBN="1-56592-051-1">
    <タイトル>TeXの作成・TeXでの作業のために</タイトル>
    <著者>ノーマン ウォルシュ</著者>
  </書籍>
  <!-- ここにはもっと多くの書籍があると思ってください... -->
</カタログ>
```



Why Python? (1/4)



- **1. Python source code is easy to read and maintain.**
 - **As a programming language, Python exhibits a remarkable clarity of expression.**
 - **Maintaining code written by another programmer becomes much easier simply because its easier to concentrate on the actual structure and algorithms of the code.**
 - **For the individual programmer, this is a nice side benefit, but for a busyness, this results in lower expenses for code maintenance.**

- `#! python3`
- `# mapIt.py - コマンドラインやクリップボードに指定した住所の地図を開く`
- `import webbrowser, sys, pyperclip`
- `if len(sys.argv) > 1:`
 - `# コマンドラインから住所を所得する`
 - `address = ''.join(sys.argv[1:])`
- `else:`
 - `# クリップボードから住所を取得する`
 - `address = pyperclip.paste()`
- `webbrowser.open('http://www.google.com/maps/place/' + address)`
住所例



Why Python? (2/4)

- **2. The interactive interpreter makes it simple to try out code fragments.**
- **Many modern high-level programming language offer interpreters, but have proved as successful as doing so as Python.**
- **Others, such as Java, do not generally offer interpreters at all. If we consider Perl, a language that is arguably very capable when used from a command line, we see that it is not equipped with a rich interpreter.**
- **With Python, the interactive interpreter provides a rich environment for executing individual statements and testing result.**



```
*googling.py - C:\Users\kagay\OneDrive\Kagayama\Prog...
File Edit Format Run Options Window Help
# python3
# googling.py - Google検索結果をいくつか開く

import requests, sys, webbrowser, bs4

print("検索文字列を入力してください。")
keyword = input()
print("googling..." + keyword + "に関するサイトを見つけるためにググっています…") # Googleページをダウンロード中に表示
res = requests.get('http://google.com/search?q=' + keyword)
# res = requests.get('http://google.com/search?q=' + keyword)

res.raise_for_status()

# 上位の検索結果のリンクを取得する
soup = bs4.BeautifulSoup(res.text)
link_elems = soup.select('.r a')

# 各結果をブラウザのタブで開く
num_open = min(5, len(link_elems))
for i in range(num_open):
    webbrowser.open('http://google.com' + link_elems[i].get('href'))

Ln: 9 Col: 0
```



Why Python? (3/4)

- **3. Python is incredibly portable, but does not restrict access to platform-specific capabilities.**
 - **The Python interpreter is one of the most portable language interpreters available.**
 - **It is known to run on platforms ranging from PDAs and other embedded systems to some of the most powerful multiprocessor platforms ever built.**
 - **It can run on more operating systems than perhaps any other interpreter.**



Why Python? (4/4)



- **4. The object-oriented features are powerful without being obscure.**
 - **Python supports object orientation without much of the syntactic overhead found in many widely used object-oriented languages (such as C++ , Java).**
 - **When combined with the excellent advantage taken by Python's standard library of a variety of common interfaces, the value of creating reusable objects is easily recognized.**

```
f2j_j2f_simple.py - C:¥U...
File Edit Format Run Options Window Help
#! python3
# f2j_j2f_simple.py

# 仏和辞書 (f2j)
f2j = {'le chien': '犬', 'le chat': '猫', 'le souris': '鼠', 'le morse': '海象'}

# -----
# これを和仏辞書 (j2f) に変換する。
# -----
j2f = {}
for french, japanese in f2j.items():
    j2f[japanese] = french
print(j2f)

# -----
# スクリプトの実行の結果
# {'犬': 'le chien', '猫': 'le chat', '鼠': 'le souris', '海象': 'le morse'}
```



Demonstration with Python

- `mapit.py`
 - Input your address, then you can see your map of your home and street view of your neighborhood.
- `googling.py`
 - Input your concerning item, then you can find 5 Web-pages automatically.
- `f2j_j2f.py`
 - Input a French-Japanese dictionary, then you can get the Japanese-French dictionary automatically.
- `kanji2suji_suji2marusuji.py`
 - Input a XML file of any article receiving from e-Gov, then you can get revised edition of XML file changed Kansuji characters into Arabic characters automatically.



DataTech

佐々木隆仁=志田大輔 『データテック –XMLルネサンス
による最強のデータ戦略–』日経BP社(2019/3/4)

- 序章 なぜ『動かないAI』が続出するのか
- 第1章 第4次産業革命とデジタルトランスフォーメーション
- 第2章 XMLルネサンス(再興)の必然
- 第3章 新たな世界を開くXMLの自動変換技術
- 第4章 XMLの自動変換技術, ポイント解説 – 第4次産業革命を導くデータの在り方
- 第5章 XMLデータ活用で業界は一変する
- 第6章 データ志向で描くデジタル国家像
- 第7章 「データの解放」へ時代が動く



序章 なぜ『動かないAI』が続出するのか

- インターネット標準のデータ形式「XML」に改めて注目すべき理由
 - W3Cは、20年前にXML標準を勧告した際、XMLの普及によって700兆円の新たな経済圏が生まれるだろうと予測した。
 - 最近のAIやビッグデータ、ブロックチェーンなどへの広がりを考慮すれば、当時の想定価値を大きく上回るインパクトをもたらすであろう。もう一度、XMLに注目すべき時機が到来したのは間違いない。
- 「データの解放」が大変革を起こし新たな経済圏への扉を開く
 - XMLの本来のゴールは、インターネット上で機械も人間も誰もがストレートに共通のデータを使用できる姿だ。
 - それを実現するのが、「XMLの自動変換技術」である。



第1章 第4次産業革命と デジタルトランスフォーメーション

- 1-1 第4次産業革命で何が起こるのか？
 - 日本はデジタル化の最後尾？ 高いポテンシャルを生かしたい
 - コア技術の間でデータを循環 これが必要の鍵となる
- 1-2 構造化データと非構造化データ
 - 説明づけられていないデータは機械が意味を理解できない
 - 活用できるのはわずか3% 残りのデータをどうするか
- 1-3 デジタルトランスフォーメーションを阻む壁
 - 良質なデータを集められないこれがAI開発者の大きな悩み
 - 革新的なサービスを生むには形式の整った構造化データが不可欠



第2章 XMLルネサンス(再興)の必然

- 2-1 そもそもXML (eXtensible Markup Language) とは何か？
 - W3Cが勧告した世界標準の規格 タグでメタデータを付与する。
 - 銀行や病院などがすでに利用中 ブロックチェーンとも親和性がある。
- 2-2 XMLが抱えていた課題
 - XMLのデザインを、素早く準備できるようにする。
 - XMLの文書を、簡単に書けるようにする。
 - XMLは、JSONが台頭したことで、終わったって？
 - MicrosoftがXMLで記述された規格をOffice2007から標準ファイル形式として採用した。
 - .doc→.docx, xls→.xlsxなど、ファイル名の拡張子の末尾に追加されたxは、まさに、XMLのxである(本書68頁)。
 - 誰でも簡単にXMLを生成できる真にオープンな環境がゴール。



第3章 新たな世界を開くXMLの自動変換技術

■ 3-1 XMLの自動変換技術に求められる4つの機能

1. OSやデバイスに縛られずブラウザ上でXMLを扱える。
2. 情報を入力するだけで素人でもXMLを生成できる。
3. あらゆるデータを自動でXMLに変換できる。
4. ビュー情報付きのデータだから機械だけでなく人間も読める。

■ 3-2 XMLデータを効率よく大量生産

- データのコスト問題を解決せずに実ビジネスには到達できない。
- XMLデータから始めるシステムは開発の期間とコストを抑えられる。



第4章 XMLの自動変換技術, ポイント解説ー

第4次産業革命を導くデータの在り方

■ Questions

1. 今あらためて「XML」に注目すべき理由とは何でしょうか？
 - EUのGDPR(EU一般データ保護規則)のデータポータビリティ件においても, 流通データのフォーマットとしては, XMLが想定されている(本書93頁)。
2. 第4次産業革命に資するデータに必要なものは何だと考えますか？
3. XMLの自動変換技術のコンセプトはどのようなものですか？
4. 従来のシステムとはどのように違うのですか？
5. XMLの自動変換技術の利用は本当に簡単なのでしょうか？
6. 他にXMLの自動変換技術の特徴はありますか？
7. データ検索でも優位な点はありますか？
8. 既存のRDBは不要になるのでしょうか？
9. XMLの自動変換技術はどういった分野での活用が期待されますか？



第5章 XMLデータ活用で業界は一変する

- 5-1 医療業界ー大きな変革のポテンシャル
 - 頭打ちの電子カルテ普及率オープンなXMLが現状を打破できる。
 - 増え続ける日本の医療費は非効率によるコスト増が一因である。
 - 患者が自身の医療データを管理問題解決の革新的な手法とは？
- 5-2 金融業界ーXML化の先に待つインパクト
 - 銀行業界は送金の電文方式を従来の固定長からMLへ移行する。
 - スマートコントラクトでデータ活用はさらに広がる。
- 5-3 法務業界ーAI連携へデータ化が緊急課題
 - 遺言書のデータをXML化して将来の財産分与も自動化へ。
 - XMLとブロックチェーンを活用した電子契約の新しいプラットフォーム。
 - AIに法令を学習させれば法案の作成も可能になる。
 - 韓国ではすべての法令・判例データをXML化する取り組みが始まっている。
 - 従来にない使い勝手の良さを目指す「法令・判例検索」サービスの開始。DBengosi.com



第6章 データ志向で描くデジタル国家像

- 6-1 「eジャパン戦略」「電子政府」の教訓
 - 官民によるデータの利活用データの相互運用性を重視
- 6-2 医療の技術とサービスを飛躍的に向上させる
 - データヘルスで医療費を削減 患者データの活用が決め手に
 - AIが多忙な医師を支援 データ共有なしには進まず
- 6-3 「書面文化」から一気に逆転を図る
 - OECD加盟35カ国中23位の日本 IT危機感をバネ、法務の効率化へ拍車
 - 政府や自治体、民間が協力し「データ駆動型社会」の実現を図る



第7章 「データの解放」へ時代が動く

- 7-1 一潮流の変化を捉えれば見えてくるチャンス
 - データの健全な流通が閉塞感を打破する。
 - 「データポータビリティ」が新たな覇者の誕生を促す。
- 7-2 データ革命の覇者となるために
 - 時代の変革を先取りしばく大な思恵を勝ち取る。
- おわりに
 - XMLは、グーテンベルクの「活版印刷機」の発明に匹敵する。



Regal Tech

佐々木隆仁『リーガルテック』アスコム(2017/12/1)

- はじめに
- 第1章 リーガルテックが法律業界を変える
- 第2章 デジタルデータで運命が変わる
- 第3章 日本の弱点をリーガルテックで克服する
- 第4章 AI, IoTがつくるリーガルテックの未来



Bibliography

■ XML

- 山田祥寛『10日でおぼえるXML入門教室』[第2版]翔泳社(2004/10/2)

■ Python

- Bill Lubanovic(斎藤康毅=長尾高弘訳)『入門Python3』オライリー・ジャパン(2015/12/1)
- Al Sweigart(相川愛三訳)『退屈なことはPythonにやらせよう』オライリー・ジャパン(2017/6/3)
- Cory Althof(清水川貴之=新木雅也訳)『独習プログラマー Python言語の基本から仕事のやり方まで』日経BP(2018/2/26)

■ Python & XML

- Christopher A. Jones, Fred L. Drake, Jr., "Python & XML", O'Reilly(2002)

■ Data Science

- Seppe vonden Broucke=Bart Baesens(トップスタジオ訳)『Pythonスクレイピングの基本と実践ーデータサイエンティストのためのWebデータ収集術』インプレス(2018/12/21)
- 佐々木隆仁=志田大輔『データテックDataTechーXMLルネサンスによる最強のデータ戦略』日経BP社(2019/3/4)

■ LegalTech

- 佐々木隆仁『リーガルテック』アスコム(2017/12/1)
- 吉峯耕平=倉持孝一郎=藤本隆三=新井幸宏「デジタル・フォレンジックの原理・実際と証拠評価のあり方」Quarterly Keiji-Bengo no.77 Spring 2014, p, 134-154 (<https://www.fss.jp/wp-content/uploads/principle-of-df2.pdf>)

