

データ収集

《学修項目》

- ◎ IoT (Internet of Things)
- IoTを支える要素技術（通信プロトコル、無線ネットワーク）
- エッジデバイス、センサーデータ
- Webクローラー、スクレイピング
- アノテーション

《キーワード》

IoT, ユビキタスコンピューティング, サイバーフィジカルシステム, Society 5.0, 通信, インターネット, センサ, エッジデバイス, データ保存, SNS, 参加型センシング, オープンデータ, ウェブスクレイピング, アノテーション

《参考文献、参考書籍》

- [1] 東京大学MIセンター公開教材 「2-3 データ収集」 《利用条件CC BY-NC-SA》
- [2] データサイエンスの考え方 社会に役立つAI×データ活用のために（オーム社）
- [3] Pythonによるあたらしいデータ分析の教科書（翔泳社）
- [4] 数理・データサイエンス・AI公開講座（放送大学）

1. IoTとは

あらまし

- IoTの潮流
- ユビキタスコンピューティング
- IoT (Internet of Things) モノのインターネット
- サイバーフィジカルシステム (CPS)
- Society 5.0

1.1 IoTの潮流 [1]

IOTとは

もともとコンピュータ間の通信に用いられるインターネットですが、それを物との通信にも使おうという試みがあります。それが物のインターネットであるIoT(Internet of Things)です。

多くの場合、物というのはセンサー（温度計や監視カメラなど、情報を取得する装置、測定器）やアクチュエータ（ロボットアームを動かすモーターや温度の調整を行う空調機など、実効行為を行う装置）のことです。こうした物に小さな通信装置をつけてインターネットに接続することで世界中のどこからでも測定器の測定結果を見たり、離れたところにいるロボットを動かしたりできるということがIoTで想定されています。

IOTの特徴

インターネットは、もともと、コンピュータ同士の通信に使うことが想定されているので、それを物（物についての通信装置）との通信でうまく使うことができるのでしょうか。また、これまでのコンピュータ同士の通信とどこが大きく変わるのでしょうか。IoTの特徴として言われていることは、以下のようなことです。

1) 数が非常に多い

既に、2019年で84億個の物と接続しているという報告があります[1]。今後ますます物との接続数が増加することが想定されます。コンピュータの数より物の数の方が圧倒的に多いので、それだけ多くの物を接続することが可能なのでしょうか。

[1] <https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016>
Access: 2021/3/2

- Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016

IOTの特徴

2) 通信機の能力が低い

インターネットに接続されるコンピュータの多くは、高性能なCPUと多くのメモリを有する装置です。ところが物につけられる通信装置の多くは、安価で小型な分、プロセッサの能力も低く、メモリも少ないです。通信速度も低いことが多いでしょう。このような通信装置で多様なアプリケーションを効率的に実現しなければなりません。

3) 低消費電力化が必要な場合が多い

センサーはいろいろな場所に設置されます。電池駆動が必要な場合も、数多く想定されます。人が行きにくい場所に設置されたセンサーについてはもちろん、電池交換が容易でも非常に数が多い場合の面倒を考えると、消費電力を低く抑え、電池寿命を長持ちさせる必要があります。

1.2 ユビキタスコンピューティング

ユビキタスコンピューティング (ubiquitous computing) は、コンピュータがいたる所に存在（遍在）し、いつでもどこでも使える状態をあらわす概念である。「あらゆる場所であらゆるモノがネットワークにつながる」ことはユビキタスネットワークと呼ばれるようになった。ユビキタスコンピューティングやユビキタスネットワークが広まった当初はおもに、移動体通信や無線などにより携帯電話や携帯情報端末（PDA）などの持ち運び可能な機器をコンピュータネットワークと接続することが想定された。

1980年代の終わりに提唱された概念である。コンピュータはどんどん小さくなり、やがて存在を意識されないぐらい生活に溶け込むとされた。

生活家電におけるユビキタスコンピューティングの例として、(1) 服のタグにチップ(RFIDタグ)を付け、洗濯かごが洗うものを自動認識、(2) 食品の画像を使った庫内の食材をスマートで確認できるスマート冷蔵庫などがある。

- JPNIC | ユビキタスコンピューティングとは
- LaunderPal | 洗濯かごが洗うものを自動認識
- アイリスオーヤマ | カメラ付き冷凍冷蔵庫

1.3 IoT (Internet of Things) モノのインターネット

「IoT」とは「Internet of Things」の頭文字を取った単語である。簡単に説明すると「身の回りのあらゆるモノがインターネットにつながる」仕組みのこととなる。

IoT : Internet of Things (モノのインターネット) ではこれまでインターネットとは無縁だったテレビやエアコンがインターネットにつながることにより、モノが相互通信し、遠隔からも認識や計測・制御などが可能となる。

人が操作してインターネットにつなぐだけではなく、モノが自らインターネットにアクセスすることがIoTの特徴である。

- インターネットアカデミー | IoT : Internet of Things（モノのインターネット）とは？
IoTの活用事例

IoTデバイスの例：RFIDタグ（引用：株式会社日立ソリューションズ・クリエイト）

RFID（Radio Frequency Identification）とは、近距離の無線通信を用いてID情報などのデータを記録した専用タグと非接触による情報のやりとりをする技術の総称である。

人を介さずに専用タグからデータを読み込んで内容を認識する自動認識技術のひとつとして知られている。

専用タグはRFタグ、もしくはRFIDタグ、ICタグなどと呼ばれる。

このタグはデータを書き込めるICチップ、アンテナ、コンデンサなどで構成されている。専用の読み込み装置であるリーダライタを使って、タグとの間で近距離無線通信が行われて情報がやりとりできる。

RFIDの活用例：交通系ICカード、高速道路のETCカード、非接触ICカード、車のスマートキーなどがある。無人レジ、商品の在庫管理や棚卸しにもRFIDは利用されている。

- 引用：株式会社日立ソリューションズ・クリエイト
- Locus Journal | RFIDタグを導入したユニックロから学ぶ他業界RFID活用のヒント

1.4 サイバーフィジカルシステム (CPS)

CPS = 現実社会からIoTを使ってビッグデータを収集、解析して現実社会へ還元する（引用：JEITA）

CPSとは、実世界（フィジカル空間）にある多様なデータをセンサーネットワーク等で収集し、サイバー空間で大規模データ処理技術等を駆使して分析／知識化を行い、そこで創出した情報／価値によって産業の活性化や社会問題の解決を図っていくこと。

テクノロジーのさらなる進化は、これまで実現できなかったデータの収集・蓄積・解析、解析結果の実世界へのフィードバックといった一連のサイクルを社会規模で可能にしている。

実世界とサイバー空間が相互連携した社会（CPS/IoT社会）においては、私たちとインターネット空間の接点はパソコンやスマートフォンといった端末に留まらず、車や家といった生活空間に広がり、収集されたデータはあらゆる分野と連携し生活をより豊かにするとともに、少子高齢化やエネルギー問題といった私たちが抱える社会的な課題の解決へも繋がっていく。

- JEITA | サイバーフィジカルシステム (CPS) とは

1.5 Society 5.0

経済発展と社会問題の解決を両立する人間中心の社会。サイバーフィジカルシステムを基盤として利用する（引用：内閣府科学技術政策）

Society 5.0とは、サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する、人間中心の社会（Society）のことである。

狩猟社会（Society 1.0）、農耕社会（Society 2.0）、工業社会（Society 3.0）、情報社会（Society 4.0）に続く、新たな社会を指すものである。

- 内閣府 | 科学技術政策 Society 5.0

2. IoTを支える要素技術

2.1 通信

あらまし

- 通信技術、通信プロトコル
- Internet接続（携帯電話回線、無線WAN、公衆無線LAN、光接続サービス）
- LAN: Local Area Network（有線LAN(GbE)、無線LAN(WiFi)、無線PAN(Bluetooth)）
- WAN: Wide Area Network（インターネット接続サービス、4G/5G携帯電話網）
- LPWA: Low Power Wide Area（低消費電力で数km以上の通信範囲を持つ無線規格。IoTデバイスをインターネットに接続する用途にも使われる）

2.1.1 通信技術、通信プロトコル、レイヤ、ヘッダ・PDU[1]

通信技術・通信プロトコル

データの収集には、通信ネットワーク、通信技術が必須です。

そして、情報を受けるには多くのルールが必要です。

例えば、AさんがBさんに情報を伝える場合、

- 英語を用いるのか、日本語なのか
- 手紙を渡すのか、伝言するのか
- 直接会うのか、誰に仲介を頼むか
- 仲介者はどのように決めるのか
- 情報を受けとったことを確認する方法はどうするのか

などを決める必要があります。通信に関わるこうしたルールを通信プロトコルと言います。

- ネットワークエンジニア | プロトコルとは

通信プロトコルレイヤ

通信プロトコルは、通常、レイヤ（層）と呼ばれる階層に分かれており、階層ごとに用いるプロトコルを決めることで通信が成立します。また、階層化することで各階層のプロトコルを適切に組み合わせることが可能になります。また、開発もプロトコルごとに別々に開発可能。最上位層をアプリケーション層、最下位層を物理層と呼びます。アプリケーション層と物理層の間にあり、どこを経由して相手先まで情報を届けるか（経路選択あるいはルーティングなどと呼ばれる）を、主に行っている層をネットワーク層と呼びます。



- ネットワークエンジニア | OSI参照モデルとは

ヘッダ

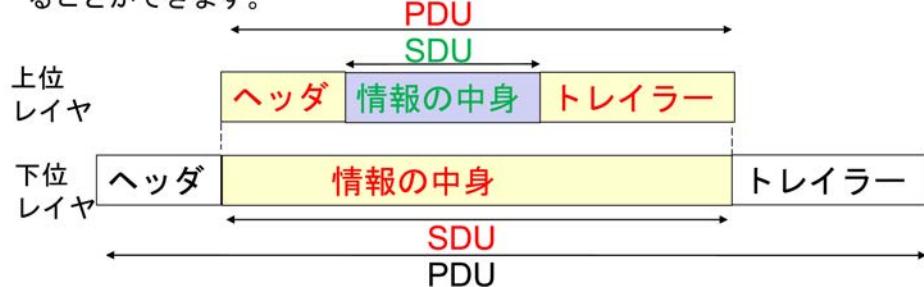
各レイヤの情報は、先頭にヘッダと最後にトレイラーと呼ばれる付加情報に挟まれて送られます。（トレイラーがないプロトコルも多いです。）ヘッダには、あて先アドレスなどの情報が格納されます。あて先アドレスは仲介者（中継ノード）のアドレスの場合や最終目的地の場合など、プロトコル、レイヤで変わります。アドレス体系もプロトコルで変わります。電話番号と郵便番号のようなものです。全体として、封筒とその中に入れた手紙のような構造になります。



- ネットワークエンジニア | IPパケット・ヘッダー

PDU

ヘッダ、トレイラーを含めた全体をプロトコルデータユニット（PDU）、情報の中身の部分をサービスデータユニット（あるいはペイロード）と呼びます。1つ上位のレイヤのPDUは、その下のレイヤのSDU（情報の中身）になるという形の入れ子構造になります。情報を送る側は、最上位のアプリケーション層から順にPDUを構成し下位層に渡します。PDUを受け取った最下位層の物理層において無線や光などの物理的な手段で実際の通信を実行します。この通信を受信した相手先は、今度は、逆に最下位層から順に情報の中身を取り出して1つずつ上位層に渡していくことでアプリケーション層で、送信側のアプリケーション層が作った情報の中身を得ることができます。



東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

7

- ネットワークエンジニア | イーサネット・フレーム

2.1.2 インターネット, HTTP, URL, DNS

インターネット

広域にまたがるコンピュータのネットワークには、多くの場合、インターネットと呼ばれるネットワークが使われています。インターネットのネットワーク層のプロトコルとしては、インターネット・プロトコル(IP)が用いられます。

IPで用いられるPDUはIPパケットと呼ばれ、そのヘッダ(IPヘッダ)にはIPアドレスと呼ばれる先アドレスが含まれます。

IPアドレスには32ビットで表現するIPv4(バージョン4)アドレスと128ビットで表現するIPv6(バージョン6)アドレスがあります。IPv4アドレスは、ほぼ使い尽くされているので、今後は、IPv6アドレスの使用が盛んになると思われます。

インターネットのネットワーク層はIPですが、それ以外の階層には多様なプロトコルを用いることができます。従って、インターネットを構成する物理的なネットワーク(物理層)として光や無線が混在することが可能になります。また、アプリケーション層としては後述するHTTPやFTPなどの多様なプロトコルを用いることができます。

東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

8

- ネットワークエンジニア | IPアドレス

アプリケーションプロトコル (HTTP)

HTTP (Hyper Text Transfer Protocol)はWebサーバーとクライアント（サーバと接続する相手）がWebコンテンツをやりとりするためのプロトコルです。Webコンテンツの記述言語であるHTML (Hyper Text Markup Language)で書かれたテキストや画像をメタデータを含めてやりとりすることができます。

クライアントのHTTPリクエストに対してサーバがHTTPレスポンスを返すことが基本です。HTTPリクエストは、画面上の特定の場所をクリックしたり、URL(後述)を指定したりすることで送出される場合が多く、HTTPメソッド、対象までのパス、などからなります。

HTTPメソッドには、「やりたい動作」などの情報が含まれます。最も良く使われる「やりたい動作」はGETで、これは「パスの指定先から取ってくる」ことを意味します。HTTPレスポンスは、リクエストが成功したか否かなどを表すstatus codeと呼ばれる回答を含むヘッダと本体（コンテンツ取得時は、取得したコンテンツ）が主たる内容になります。

HTTPでは、通信内容が暗号化されていないため、セキュリティ上の懸念があることから、最近では、暗号化を行うHTTPであるHTTPSの利用が増えてきています。

東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

18

- ネットワークエンジニア | HTTP

URL、DNS

HTTPをはじめアプリケーションプロトコルの中には、コンテンツなどの位置を一種のアドレスであるURL(universal resource locator)で指定するものがあります。URLの構造はプロトコル、コンテンツのあるコンピュータのドメイン名、パス名からなります。

https://www.u-tokyo.ac.jp/ja/index.html

一方、指定された位置に到達することはアプリケーション層ではなくネットワーク層の機能です。しかし、インターネットの場合のネットワーク層のプロトコルであるIPはURLを解釈できないため、URL（から得たドメイン名）をIPアドレスに変換する必要があります。その仕組みはDNS(domain name system)と呼ばれます。DNSは階層化され、上位DNSは下位DNSを知っており、該当ドメインDNSが所属ホストの情報を持っているという仕組みになっています。そのため、世界中のドメインとIPアドレスを対応づけることができます。

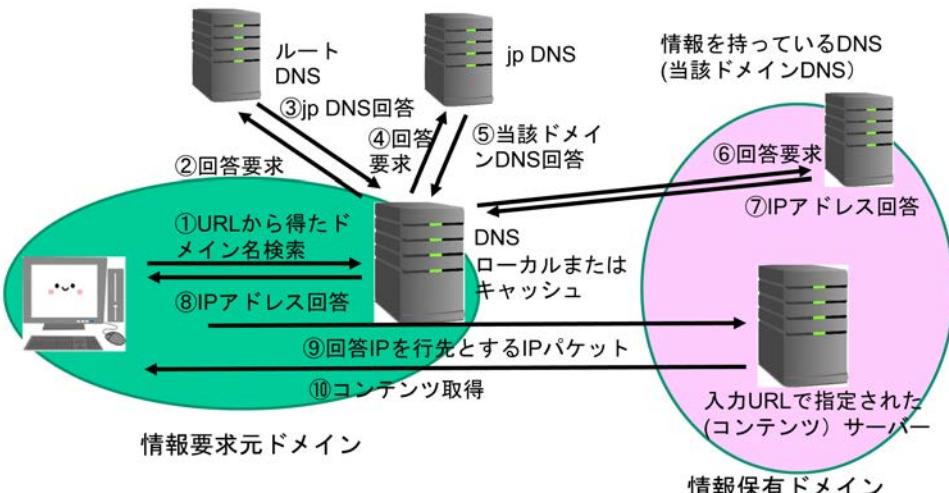
東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

19

- RFC1738 - Uniform Resource Locators (URL)
- ネットワークエンジニア | DNS

アドレス変換から見たコンテンツ取得までの流れ

jp傘下のドメインの場合



東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

20

Pythonを使用したDNSクエリ、HTTPサーバへのGETコマンド送信の例

```
In [ ]: # dnsPYTHONモジュールのインストール  
!pip install dnsPYTHON
```

```
In [ ]: import dns.resolver  
  
answers = dns.resolver.query('gmail.com', 'MX') # GMailメールサーバのMTAを調べる  
for rdata in answers:  
    print('Host', rdata.exchange, 'has preference', rdata.preference) # MXレコード
```

```
In [ ]: # HTTPプロトコル経由で HTMLファイルをサーバからダウンロードする  
  
import urllib.request  
  
# URLを指定すると urllib経由でOS機能(レゾルバ)によってDNS問い合わせを行い,  
# GETリクエストを発行してレスポンスをdataへ格納できる(たった1行!)  
data = urllib.request.urlopen("https://www.google.com/")  
  
html = data.read()  
print(html)  
data.close()
```

2.1.3 センサネットワーク

センサーネットワークとは

IoTでは、センサーに通信装置を付けて通信することで様々な状況を測定、監視できることが期待されています。IoTでは、センサーにつけた通信装置がIPアドレスを持ち、IPを用いて通信することになります^(*)。しかしながら、これ以外の方法で通信装置付きセンサーと通信し、様々な状況の測定、監視は可能です。一般的に、こうしたセンサーとの通信を行うネットワークをセンサーネットワークと呼びます。

IoTも一種のセンサーネットワークとして捉えることができるので、IoTの特徴、「膨大な数」「低能力通信ノード」「低消費電力要求」は、そのままセンサーネットワークの特徴となります。そしてこれらの特徴は、そのまま、センサーネットワークが抱える課題となります。

センサーが計測したデータ（センサーデータ）は、センサーネットワークにより収集されます。

(*) そうでない場合も、IoTと呼んでいるケースもあるようです。

センサーネットワークアーキテクチャ

センサーネットワークの構成は、ゲートウェイ装置(GW)の介在あり、無し、で大別されます。以下の図で、ネットワーク(NW)の典型例はインターネット、あるいは、携帯電話網です。



(a) GWが無い場合



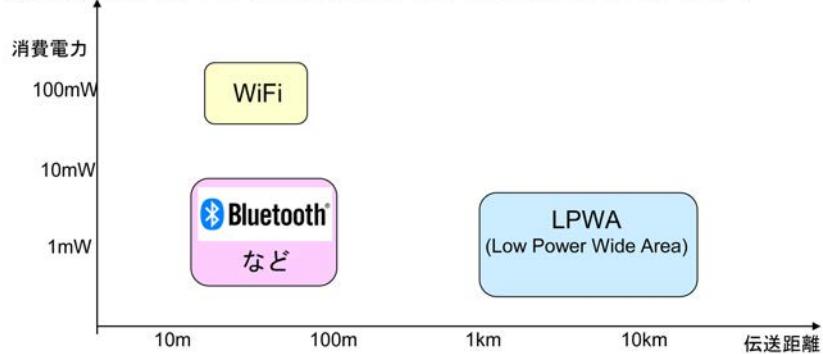
(b) GWがある場合

- (a)の典型例がセンサーについていた通信機がIPアドレスを使う場合です。
- (b)の場合の「GW-NW-サーバ」部分は、通常のコンピュータ間接続などと同様ですので、次ページ以降では、センサーアクセスNW部分を中心を見てみましょう。

2.1.4 無線ネットワーク規格

センサーアクセスマルチホップ用無線通信規格

センサーアクセスマルチホップには、イーサネットなどの有線ネットワークも使われますが、多くの場合、無線のネットワークが使用され、その用途に応じて、多くの規格があります。その特徴を消費電力と通信距離で分類したものが以下の図になります。WiFiやBluetoothは皆さんも聞いたことがあるでしょう。それに加えて、最近では、LPWAと呼ばれる通信規格に属するものが提案され、利用され始めています。



横谷哲也、IoTと通信ネットワーク技術、電子情報通信学会誌、102、5、pp. 383-387、2019の図4
をもとに作成

東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

14

- IEEE 802.11 WIRELESS LOCAL AREA NETWORKS
- Bluetooth コア・スペック 5.3
- 株式会社バディネット | LPWAの規格一覧と比較表

センサーアクセスマルチホップ用無線通信規格

LPWAグループの通信規格は低消費電力で広い範囲(数km)をカバーすることを目的とします。通常、速度は遅いため、大量の情報を一度に送出するのではなく、少しずつ出すような通信に向いています。低消費電力であるため、電池で動作し、電池交換なしで長く使われるような状況に適しています。2012年ごろからLoRa、SIGFOXなどの規格が商用利用されるようになりました。また、携帯電話網のLTE(Long Time Evolution)をセンサーネットワークに用いるためにNB-IoT(Narrow Band-IoT)が開発されました。LTEは高速広帯域ですが上記のような用途には消費電力が大きすぎ高価格です。NB-IoTにより低速狭帯域ですが低消費電力、低価格を実現しています。NB-IoTの利用にはライセンスが必要ですので、携帯電話会社から提供される通信装置を使う必要がありますが、干渉などが管理されています。一方、LoRa、SIGFOXはアンライセンスですのでWiFiのように自由に設置可能ですが、干渉などによる性能低下の懸念は生じます。

東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

15

- 株式会社バディネット | LPWAの規格一覧と比較表

センサーアクセスネットワーク用無線通信規格 近距離無線

Bluetoothは非常に良く使われています。2019年には42億台が出荷されました[1]。主には、スマホやパソコンとイヤホンやマウスなどの周辺機器の接続が典型的な使用例です。ペアリングと呼ばれる動作によって2つの装置間が結ばれます。片方がマスター、もう一方がスレーブとなり、スレーブはマスターの指示により動作するだけで、スレーブ同士の直接通信はできません[2]。パソコンやスマホが通常マスターになります。それらの機器はGWになり得ますので、これでマウスなどのセンサーとGW間のセンサーアクセスネットワークができたことになります。Bluetoothの新しい規格では中継機能が追加され、センサー同士でネットワークが構成可能になっています。



[1] https://www.bluetooth.com/wp-content/uploads/2020/03/BMU_2020-JPN.pdf

Access: 2021/3/2

[2] アンドリュー・S・タネンbaum／デイビッド・S・ウェザロール、コンピュータネットワーク、日経BP社、2013

- [Bluetooth コア・スペック 5.3](#)
- [bluetooth.com | Bluetooth市場動向](#)

2.2 センサ

あらまし

- カメラ, マイク, 赤外線
- 位置情報
- 加速度, 角速度, 地磁気, 気圧
- 湿度, 温度, 二酸化炭素濃度
- 脈拍, 心電図, 血中酸素濃度, 血圧
- LiDAR など

2.2.1 カメラ, マイク, 赤外線

監視カメラ,赤外線センサの利用に関して ([引用：経済産業省](#))

撮影機器の著しい発達に伴い、カメラ（又はそれに準じる機器）で取得することの可能なデータが多岐に亘り、それらデータを利活用する目的も多様化している。これにより、生活者はカメラ撮影によって取得された画像がどのような目的で取得され、どのような利活用をされているかがさらに把握しにくくなっている側面もあることから、カメラ画像利活用に対する生活者の受容性を担保するため、カメラ画像の存在と画像の利用目的を明示する等が必要である。

事業者が、カメラ画像等、生活者の情報を取り扱う場合には、個人情報保護法を遵守するだけでなく、生活者のプライバシーや肖像権が私法上も保護されており、その侵害に対して生活者による損害賠償請求や差止請求が認めら

れることを認識し、生活者的人格的な権利・利益を損なうことのないよう、十分な配慮をすることが求められる。

- [経済産業省 | カメラ画像利活用 ガイドブック](#)

2.2.2 位置情報

測位の基本

- 正確な位置が分かっている基準点を複数使う
- 基準点からの距離を測定するデバイス

GNSS (Global Navigation Satellite System), GPS (全地球測位衛星システム) とは ([引用：国土地理院](#))

GNSS(Global Navigation Satellite System / 全球測位衛星システム)は、米国のGPS、日本の準天頂衛星（QZSS）、ロシアのGLONASS、欧州連合のGalileo等の衛星測位システムの総称です。

GPS (Global Positioning System) は、米国によって、航空機・船舶等の航法支援用として開発されたシステムです。このシステムは、上空約2万kmを周回するGPS衛星（6軌道面に30個配置）、GPS衛星の追跡と管制を行う管制局、測位を行うための利用者の受信機で構成されています。

航空機・船舶等では、4個以上のGPS衛星からの距離を同時に知ることにより、自分の位置等を決定します。GPS衛星からの距離は、GPS衛星から発信された電波が受信機に到達するまでに要した時間から求めます。衛星から発信される電波には、衛星の軌道情報・原子時計の正確な時間情報などが含まれています。

- [国土地理院 | GNSS\(Global Navigation Satellite System / 全球測位衛星システム\) とは](#)

Wi-Fi測位

市中に多数設置されたWi-Fi基地局からの電波で測位する。WiFi基地局は市中に大量に設置されており、事前にIDと位置の情報を集めてデータベース化する。距離の測定は電波の強度などを用いる。電波の強弱でよその距離を推定できる。不安定なため、距離の推定には使用しないこともある

([引用：センサイト・プロジェクト](#))

Wi-Fi測位技術はその測位方式の違いにより2種類に大別できる。1つ目は、多辺測位(multilateration)・多角測位(multiangulation)である。Wi-Fi通信で得た距離や角度を用いて、いわゆる三辺測量・三角測量を行う。2つ目は、位置指紋測位(fingerprinting)である。あらかじめ測位エリアの各所で何らかの物理量を測定しておき、測位対象端末で測定した物理量と近い物理量となる位置を探索することで測位を行う。

[センサイト・プロジェクト | 屋内Wi-Fi測位の基本と最前線\(1\)](#)

2.2.3 加速度、角速度、地磁気、気圧

アプリケーション

- 角速度：物理的な動作をUIに応用する。行動推定
- 地磁気センサ：方角を捉えることが可能
- 気圧計：階段の上下移動なども検出できる

MEMS (Micro Electro Mechanical Systems) (引用：ビジネス+IT)

MEMSは「Micro Electro Mechanical Systems」の頭文字を取ったものであり「メムス」と読む。

シリコンウェハーなどの上に、電子回路やセンサ、機械的に動くアクチュエーターなどを作りこんだ部品。半導体集積回路の製作技術などを利用している。

MEMSは、電子回路に加え、センサやアクチュエーターなどの可動部を動かすための立体構造を備えている。

たとえば、スマートフォンでは特定の電波を選択するためのBAW（Bulk Acoustic Wave／弾性波）フィルタ、加速度センサ、ジャイロセンサ、マイクロフォン、気圧センサなどの複数のMEMSを利用しているが、それらを統合的に制御するのが半導体集積回路の役目になる。

- ビジネス+IT | MEMS（メムス）を簡単に解説。マイクやセンサに活用の技術、LiDARなど将来性は？

2.2.4 湿度、温度、二酸化炭素濃度

アプリケーション

- 部屋の環境が基準を満たしているか
- 農作物に水は足りているか
- 火災警報

環境センサ (引用：株式会社クローネ)

環境センサとは、水や空気、土壤の状態を定量的に把握するために用いられる装置です。環境センサによって測定された情報は、検査、記録、保存などの目的でも利用可能です。また、センサと通信機器を使うことで、現地にいなくても遠隔地から状況を把握できるようになります。

温度・湿度センサ、風速・風向センサ、CO₂センサ、PM2.5センサ、ガスセンサ、濁度センサ、水質センサ、放射線センサ、感雨センサ、土壤センサなどがある。

- 株式会社クローネ | 環境センサとは？10種類のセンサと用途を解説

2.2.5 脈拍、心電図、血中酸素濃度、血圧

アプリケーション

- スマートウォッチにも内蔵されている
- 日常生活での体調の急変などの検出が期待されている

バイタルサインセンシング技術動向（引用：日経XTECH）

病院中心のケアから、地域や在宅中心のケアへーー。医療・ヘルスケアにこうしたパラダイムシフトが起こりつつある中、日常の生活シーンにおいてバイタルサインを測定し、疾患の早期発見や重症化予防などにつなげる技術（バイタルサインセンシング技術）への関心が高まっている。心拍/脈拍から血圧、心電、血中酸素まで、その対象は多岐にわたる。携帯機器やウェアラブル機器、クラウドコンピューティングなどの登場により、これらのバイタルサインを簡便にセンシングし解析できるようになってきた。技術とビジネスの両面で大きな成長が見込まれる分野である。

- [日経XTECH | バイタルサインセンシング技術に関する特許分析と事業化動向](#)

2.2.6 LiDAR

センサ原理

- 光が反射して戻ってくるまでの時間で物体形状を測定する
- Light Detection And Ranging
- レーザ光を回転させながら、光の届く範囲を探索する

LiDARの用途例 - 自動車の高度自動運転システム（引用：ROHM）

自動車の高度自動運転システムのうち、完全自動運転（レベル5）の実現にはLiDAR技術は必要不可欠である。

民生分野ではロボット掃除機やゴルフ測距計、産業機器分野では自動搬送車(AGV)やサービスロボットなど、人や物を高精度に検知する用途に利用されている。

高度自動運転システム ADAS（先進運転支援システム：例えば自動ブレーキやレーンキープなど）では、カメラ方式とミリ波レーダー方式の組み合わせが主流です。3種類のセンサはそれぞれ環境条件によって長所短所があるため、自動車の自動運転化に向けては、これらにLiDARを加えた3方式の組み合わせが必要と言われている。

- [ROHM | LiDAR（ライダー）技術](#)

2.3 エッジデバイス

2.3.1 エッジデバイス、エッジコンピューティング

エッジデバイス、エッジコンピューティング

クラウドコンピューティングが遠隔地にあるクラウド(2-1参照)上のコンピュータを利用して計算を実行するのに対して、センサーなどデータを発生させるデバイス（エッジデバイス）に比較的近い位置にあるコンピュータで計算を行うことをエッジコンピューティングと言います。このコンピュータとエッジデバイスが一体化したケースもあります。エッジコンピューティングはクラウドコンピューティングに比し通信遅延を抑制することができるため、プラントの制御や車両の自動運転などの遅延に対する要求が厳しい処理に適しています。

2.3.2 IoT端末になりうるデバイス

- スマートフォン、スマートウォッチ
- 家電
- スマートスピーカー
- カーナビ、ドラレコ
- センサデバイス

参考：[Raspberry Pi \(ラズパイ\) を使ってできること 使い方まとめ – あなたの制作意欲を刺激する作品例64選【2020年11月版】](#)

2.4 記録したデータの保存

ダイレクト型

- 各種センサ端末からインターネット上のクラウドにアップロード
- 個々の機器は、携帯電話網などに接続する

クローズド型

- Internet接続しない
- 工場内機器の制御など、その場で全ての処理を完結する
- Internet側からの攻撃可能性が低い

エッジデバイス型

- 各地で集計した後、統計データのみをアップロード

オフライン・巡回収集型

- センサ端末にデータを蓄積

- 定期的に巡回する作業者が、無線でデータ収集

3. ウェブからの情報収集

3.1 SNSからの情報収集

SNSからデータを集めることのメリット

- SNSは情報の宝庫
- 日々の生活スタイルが蓄積されている
- 観光地での導線を追跡できる
- 商品・製品、サービスを使った感想やレビューが蓄積されている
- 困っていること、悩み事、質問が蓄積されている

[参考：マーケティングでも使える！SNS情報自動収集ツール5選](#)

3.2 参加型センシング

アプリケーション

- ゲーミフィケーション
- ユーザの興味や少しの報酬と引き換えに情報収集する
- 場所毎のリアルタイムな天気を収集する
- 鉄道の運行情報の集中

ゲーミフィケーションで検討すべき要素（[引用：JCS](#)）

ゲーミフィケーションをビジネス等に応用するというと、ポイントやレベルの要素を入れればいいと思われるかもしれない。ゲーミフィケーションとはそうした表面的なものだけでなく、利用者との関係性を強化して、ゲームのように夢中にさせていく一連の行動デザインである。ゲーミフィケーション施策を試す際には、以下に挙げる必須の要素をおさえておく。

目的、クエスト、報酬、可視化

- [JCS | ゲーミフィケーションとは？必須要素と事例＆バトルテストの解説！](#)

3.3 オープンデータの取組み

オープンデータの定義

- 営利目的 非営利目的を問わず二次利用可能なルールが適用されたもの
- 機械判読に適したもの
- 無償で利用できるもの

[\(引用：高度情報通信ネットワーク社会推進戦略本部 オープンデータ基本指針 H29.5.30\)](#)

オープンデータ

- ウェブには、行政・研究機関、企業、個人などの第三者が提供・発信するさまざまなデータがあります。
- 例えば、政府統計のポータルサイトである「e-Stat」では、国内の様々な統計データを検索して入手することができます。
 - e-Stat : <https://www.e-stat.go.jp/>
- 機械（コンピュータ）の読み取りに適したデータ形式で、二次利用が可能な利用ルールで公開されたデータをオープンデータと呼びます。
- 政府のカタログサイト「DATA.GO.JP」では、二次利用可能な様々なオープンデータを検索して入手することができます。
 - DATA.GO.JP: <https://www.data.go.jp/>

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

25

- 総務省統計局 | s-Stat 統計で見る日本
- 政府のカタログサイト DATA.GO.JP

オープンデータの例

- 気象情報
- 政府統計、経済動向、消費動向指数
- 公共交通の時刻表 運行バス法 乗降客数
- インフラの整備状況

オープンデータ

e-Statのサイト

The screenshot shows the homepage of the e-Stat website. At the top, there's a header with the e-Stat logo, a search bar, and links for "統計で見る日本" (Statistics to See Japan), "お問い合わせ" (Contact), "ヘルプ" (Help), "English", "ログイン" (Login), and "新規登録" (New Registration). Below the header, there are two main sections: "統計データを探す" (Search for statistical data) and "統計データを活用する" (Use statistical data). The "統計データを探す" section includes buttons for "すべて" (All), "分野" (Field), and "組織" (Organization), along with a search bar and a "検索" (Search) button. The "統計データを活用する" section includes buttons for "グラフ" (Graph), "時系列表" (Time series table), "地図" (Map), and "地域" (Region). To the right, there are several informational boxes: "利用ガイド" (User guide), "統計データの高度利用" (Advanced use of statistical data), "ミクロデータの利用" (Use of microdata), "開発者向け" (For developers), and "統計関連情報" (Information related to statistics).

出典：政府統計の総合窓口(e-Stat)
(<https://www.e-stat.go.jp/>)

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

26

オープンデータ

- 教育用標準データセットは、データ分析のための汎用素材として、e-Statの統計データを元に作成され公開されているデータセットです。
 - 教育用標準データセット：<https://www.nstac.go.jp/SSDSE/>
- データセットは表計算ソフト形式またはCSV形式で公開されており、以下のデータが含まれています。
 - A. 市区町村別データ
 - 1741市区町村、125項目の統計データ
 - B. 都道府県別・時系列データ
 - 47都道府県、12年分、107項目の統計データ
 - C. 都道府県庁所在市別・家計消費データ
 - 都道府県庁所在市、227項目の統計データ

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

27

- [独立行政法人統計センター | SSDSE（教育用標準データセット）](#)

オープンデータ

教育用標準データセット A. 市町村別データ（表計算ソフト形式）

	A	B	C	D	E	F	G	H	I	J	K	L
1	code	prefecture	municipality	A1101	A110101	A110102	A1102	A110201	A110202	A1301	A130101	A130102
2	year	年度	年度		2015	2015	2015	2015	2015	2015	2015	2015
3	地域コード	都道府県	市区町村	総人口	総人口 (男)	総人口 (女)	日本人人口	日本人人口 (男)	日本人人口 (女)	15歳未満人口	15歳未満人口 (男)	15歳未満人口 (女)

<https://www.nstac.go.jp/SSDSE/data/2020/SSDSE-2020A.xlsx>

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

28

- [SSDSE-市区町村（SSDSE-A）データセット CSVダウンロード](#)
- [SSDSE-市区町村（SSDSE-A）データセット Excelダウンロード](#)

データの形式

- ・ オープンデータなどで公開されているデータには以下のような代表的な形式があります。
 - ・ 表計算ソフト形式
 - ・ 表計算ソフトで読み込み可能な形式
 - ・ CSV形式
 - ・ データの値をカンマ (,) で区切って表したもの
 - ・ 拡張子は.csv
 - ・ XML形式
 - ・ データの値をその種類を表すタグとともに表したもの
 - ・ 拡張子は.xml
 - ・ JSON形式
 - ・ JavaScriptのオブジェクト表記法を元にデータを記述したもの
 - ・ 拡張子は.json
- ・ XML形式やJSON形式はWeb APIを用いてウェブサービスからデータを取得する際にも利用されます。

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

29

Pythonを使用したJSONオープンデータのHTTP/API経由での取得

```
In [ ]: # HTTPプロトコル経由で サーバからJSONで取得する
# 参照元:京都市オープンデータ https://data.city.kyoto.lg.jp/riyou/api

import json
import urllib.request

# API用京都市の施設情報(平成28年11月9日現在)リソースID番号
json = 'https://data.city.kyoto.lg.jp/API/action/datastore/search.json?resou
data = urllib.request.urlopen(json)
json_data = data.read()
print(json_data)

# JSONを整形した結果を(人間が)見たい場合は以下(表示が長いので注意)
##pdata = json.loads(json_data)
##print(json.dumps(pdata, indent=2))

data.close()
```

オープンデータの意義

- ・ 公開側：あまりコストをかけない
- ・ サービス提供者：便利なサービスを低成本で提供
- ・ 利用者側：誰かがうまく使ってくれるかもしれない

ウェブからのデータ収集における留意点

- データの信ぴょう性
 - ウェブに公開されているデータを収集する際には、元データの公開元、元データ自体の収集方法や内容、などの検証を十分に行い、データの信ぴょう性を確認する必要があります。
- バイアス
 - 収集したデータにはバイアス（偏り）が含まれる可能性があることに留意する必要があります。
 - 選択バイアス：データを集める際に観測したものと観測しなかったものの間の性質の差によって生じるバイアス
 - 情報バイアス：観測者の先入観や観測対象の過少申告や過剰反応によって生じるバイアス
- 個人情報の扱い
 - データを収集する際には、個人情報の扱いに十分に留意する必要があります。収集したデータが個人情報を含む場合は、あらかじめ利用目的を公表しておくか、または取得後速やかに利用目的を本人に知らせなければいけません。

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

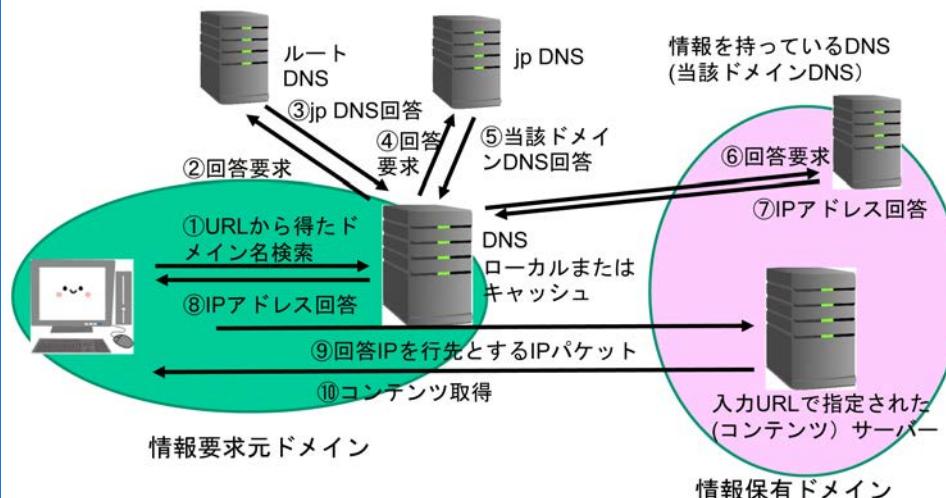
30

3.4 データ取得・収集の自動化

3.4.1 インターネット上からのコンテンツ収集、FTP、SSH（セキュアトンネル）

アドレス変換から見たコンテンツ取得までの流れ

jp傘下のドメインの場合



東京大学 数理・情報教育研究センター 斎藤洋 2021 CC BY-NC-SA

20

アプリケーションプロトコル（FTP）

FTP (File Transfer Protocol)は、クライアント（サーバを使う人）とサーバの間でファイルをやりとりするためのプロトコルです。FTPは制御用のコネクションと実際にデータ（ファイル）を転送するためのコネクションを別に用意します。2つの見えない線をクライアントとサーバ間に用意すると思ってください。制御用コネクションを通じて、データ転送用のポート番号と呼ばれる通信のための番号をやりとりします。決定されたポート番号を用いて、データをデータ転送用コネクションで転送します。

FTPは、認証のためのパスワードなどの情報やデータ自体が暗号化されずに送られるため、セキュリティ上の懸念があります。その場合は、次ページで勉強するSSHなどの仕組みを利用するなど、FTPを改良したプロトコルが使われ始めています。

- ネットワークエンジニア | FTP

アプリケーションプロトコル（SSH）

SSH (Secure Shell)は、サーバなどネットワークに接続している機器を離れたところにいるサーバ管理者などが安全に操作するためのプロトコルです。悪意をもった人がサーバにloginしてしまうと非常に影響が大きいので、それを防ぐため、通信はすべて暗号化された状態で行われます。また、認証も、パスワード認証もありますが、より堅牢な公開鍵認証（2-6参照）が推奨されています。遠隔にあるコンピュータを操作するアプリケーションプロトコルとしては、これまでtelnetが多く使用されてきましたが、SSHの利用により安全性が大幅に向上了っています。

認証後、サーバ管理者が直接使っているPCから、遠隔にあるサーバに、sshコマンドを用いて接続して、loginして、サーバでの操作を行うという使い方が一般的です。

- ネットワークエンジニア | SSH

ウェブクローラ、スクレイピング

- ウェブサイトから自動的にダウンロードする
- 専用のアプリを使うか収集用のプログラムを作成する
- ウェブブラウザを自動操作するようなツールも開発されている
- 定期的に繰り返し行う場合には、ウェブクローリングを行う

ウェブクローラ

- ・ 一般にウェブブラウザは以下のような流れでウェブページを取得しています。
 - ・ ウェブブラウザのようなクライアントはウェブサーバへhttpリクエストをおくる
 - ・ この時、http://から始まるURL (universal resource locator) を指定する
 - ・ ウェブサーバはこのリクエストに答えてURLで指定されたコンテンツ (HTMLで記述されたファイル) を返す
- ・ **ウェブクローラ**は、ウェブブラウザが行うようなウェブサーバへのコンテンツのリクエストと受信をウェブのリンクを辿りながら逐次的に行うことで自動的にウェブのコンテンツを取得し蓄積するプログラムです。
 - ・ ウェブクローラはボットやスパイダーとも呼ばれます。
- ・ 検索エンジンではウェブクローラを用いて膨大な数のウェブページを自動で収集し索引付け（インデキシング）を行い管理しています。

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

31

ウェブクローラ

Pythonのモジュール（urllib）を用いたウェブページ取得の例

```
import urllib
取得するウェブページのURLを指定
response = urllib.request.urlopen("https://www.u-tokyo.ac.jp/en/")
print(response.getcode()) # HTTPステータスコードの表示
print(response.info()) # HTTPヘッダの表示
print(response.read()) # コンテンツの表示
response.close()

200
Date: Tue, 30 Mar 2021 07:03:08 GMT
Server: Apache
Accept-Ranges: bytes
Access-Control-Allow-Origin: http://cdn.pr.u-tokyo.ac.jp
Access-Control-Request-Headers: *
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html

b'<!DOCTYPE html>\r\n<html lang="en">\r\n  <head>\r\n    <meta charset="UTF-8">\r\n    <meta http-equiv="X-UA-Compatible" content="IE=edge">\r\n    <title>The University of Tokyo</title>\r\n    <meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=2,minimum-scale=1" user-scalable="yes">\r\n    <meta name="description" content="The official website of the University of Tokyo. Features a general introduction to the University, its research and international activities, admissions and other information.">\r\n    <meta name="keywords" content="The University of Tokyo,UTokyo,\xe6\x9d\xbb1\xe4\xba\xac\xe5\x9a\x7\xe5\xad\xaa,\xe6\x96\x99">

```

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

32

In []: # Pythonのモジュール（urllib）を用いたWebページ取得の例(スライド#32掲載のもの)

```
import urllib.request

response = urllib.request.urlopen("https://www.u-tokyo.ac.jp/en/") # 取得する
print(response.getcode()) # HTTPステータスコードの表示
print(response.info()) # HTTPヘッダの表示
print(response.read()) # コンテンツの表示
response.close()
```

ウェブスクレイピング

- ・ ウェブページ（一般にはHTMLで記述されたファイル）から情報を抽出することをウェブスクレイピングと呼びます。
- ・ ウェブクローラなどで自動で収集したウェブページからウェブスクレイピングにより情報の抽出を行うことで、ウェブから自動的にデータを収集することができます。
- ・ ウェブスクレイピングでは非構造的なウェブページから情報を抽出し、データ分析に利用可能な構造的なデータに整理します。
- ・ ウェブクローリング・スクレイピングを行う際はウェブサーバに負荷がかかるないように十分に注意する必要があります。また、サイトによってはウェブクローリング・スクレイピングを禁止している（代わりにWeb APIの利用を求める）こともあるため、事前にサイトの規約をよく確認しておく必要があります。

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

33

ウェブスクレイピング

Pythonのモジュール（Beautiful Soup）を用いたウェブスクレイピングの例

```
import urllib
from bs4 import BeautifulSoup

response = urllib.request.urlopen('https://www.u-tokyo.ac.jp/en/')
soup = BeautifulSoup(response)
response.close()
print(soup.title.text) # ウェブページのtitleタグのテキストを表示
```

The University of Tokyo 実行結果

https://www.u-tokyo.ac.jp/en/

スクレイピングにより対象ウェブページ
のタイトル情報を抽出

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=2, minimum-scale=1, user-scalable=yes" />
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
<meta name="format-detection" content="telephone=no" />
<meta name="msapplication-tap-highlight" content="no" />
<meta name="description" />
<meta content="The University of Tokyo, UTokyo, 東京大学, 東大, Todai, Japanese University" name="keywords" />
<meta content="The University of Tokyo" name="copyright" />
<link href="/content/400132641.ico" rel="shortcut icon" type="image/x-icon" />
<link href="/content/400130668.png" rel="apple-touch-icon" sizes="180x180" />
<link href="/content/400132625.png" rel="icon" sizes="192x192" type="image/png" />
<meta content="The University of Tokyo" name="og:title" />

ウェブページのタイトル情報

Tokyo. Features a general introduction, activities, admissions and other information.

スクレイピングの対象ウェブページのHTMLソース

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

34

In []: # Pythonのモジュール（Beautiful Soup）を用いたWebスクレイピングの例(スライド#34掲載のもの)

```
import urllib
from bs4 import BeautifulSoup

response = urllib.request.urlopen("https://www.u-tokyo.ac.jp/en/") # 取得する
soup = BeautifulSoup(response) # レスポンスをBSオブジェクトに格納
response.close()

print(soup.title.text) # レスポンス中のタイトル情報を抽出
```

In []: # retryモジュールをインストール
参考:https://memo.techack.net/python/retry.html

```
!pip install retry
```

```
In [ ]: # (参考情報) 【コード解説】PythonでSUUMOの賃貸物件情報をスクレイピング
# https://myfrankblog.com/scraping-suumo-website/

from retry import retry
import requests
from bs4 import BeautifulSoup
import pandas as pd

# 東京23区
base_url = "https://suumo.jp/jj/chintai/ichiran/FR301FC001/?ar=030&bs=040&ta=050"

@retry(tries=3, delay=1, backoff=1)
def get_html(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.content, "html.parser")
    return soup

all_data = []
max_page = 3 #クロールする最大ページ数を指定する.東京23区だと50ページ,全部だと1697ページもある

for page in range(1, max_page+1):
    # define url
    url = base_url.format(page)

    # get html
    soup = get_html(url)

    # extract all items
    items = soup.findAll("div", {"class": "cassetteitem"})
    print("page", page, "items", len(items))

    # process each item
    for item in items:
        stations = item.findAll("div", {"class": "cassetteitem_detail-text"})

        # process each station
        for station in stations:
            # define variable
            base_data = {}

            # collect base information
            base_data["名称"] = item.find("div", {"class": "cassetteitem_content-title"})
            base_data["カテゴリー"] = item.find("div", {"class": "cassetteitem_content-type"})
            base_data["アドレス"] = item.find("li", {"class": "cassetteitem_detail-content-item"})
            base_data["アクセス"] = station.getText().strip()
            base_data["築年数"] = item.find("li", {"class": "cassetteitem_detail-condition-item"})
            base_data["構造"] = item.find("li", {"class": "cassetteitem_detail-condition-item"})

            # process for each room
            tbodys = item.find("table", {"class": "cassetteitem_other"}).findAll("tbody")

            for tbody in tbodys:
                data = base_data.copy()

                data["階数"] = tbody.findAll("td")[2].getText().strip()

                data["家賃"] = tbody.findAll("td")[3].findAll("li")[0].getText().strip()
                data["管理費"] = tbody.findAll("td")[3].findAll("li")[1].getText().strip()

                all_data.append(data)
```

```

        data["敷金"] = tbody.findAll("td")[4].findAll("li")[0].getText().str
        data["礼金"] = tbody.findAll("td")[4].findAll("li")[1].getText().str

        data["間取り"] = tbody.findAll("td")[5].findAll("li")[0].getText().str
        data["面積"] = tbody.findAll("td")[5].findAll("li")[1].getText().str

        data["URL"] = "https://suumo.jp" + tbody.findAll("td")[8].find("a").get("href")

    all_data.append(data)

# convert to dataframe
df = pd.DataFrame(all_data)

```

```
In [ ]: # dataframeをきれいに表示して,オンデマンド・ソーティングをしてみる
import pandas as pd
from IPython.display import display

display(df)
```

ウェブAPI

- API (Application Programming INterface) 人間ではなく、プログラムにデータ提供する仕組み
- スクレイピングするよりも効率的になる
- 検索サイト、SNS、通販サイトの多くがAPIを提供している

クライアント技術 (API)

API (Application Programming Interface)は、あるソフトウェア基盤上に第3者がアプリケーションプログラムを構築するための切り口のことです。この切り口に合わせてプログラムを作れば、この基盤が所定の動作をします。この結果、多くの第三者が(APIに準拠する限り)基盤の中身の詳細を知らなくとも自由にアプリケーションを開発できますし、基盤提供者は多くのアプリケーションに使ってもらうことができます。実際のAPIはアプリケーションプログラムを書くためのルール集です。コマンド集あるいはファイル群で提供される場合もあります。

APIを提供する基盤には多くのものがあります。AmazonやGoogleなど多くのプラットフォーム事業者は自社のWebサービスを使ってもらうためにAPIを提供しています。



クライアント技術 (SDK)

SDK (Software Development Kit)は特定の基盤上のソフトウェア開発のためのツール類 (APIに関するものを含む、マニュアルやサンプルコード、開発支援用プログラムなど) を言います。通常、Microsoftなどのコンピュータオペレーティングシステム事業者、プログラミング言語メーカー、ハードウェア基盤事業者によって提供されます。SDKを利用することで、当該基盤を利用したソフトウェアの開発が容易になり、提供者にとっては、その基盤のユーザが増えることになります。

データのアノテーション

- 注釈などのメタデータを紐付ける作業
- 生データでは何を意味しているのかわからないので、周辺データ等から意味を推定する

アノテーション

- データに対してそれがどのようなデータであるかを示す情報を付加することを一般にアノテーションと呼びます。この時、付加される情報をタグまたはメタデータと呼びます。
 - XML形式のデータはXMLのタグによってアノテーションされたデータです。また、ウェブページは一般にHTMLのタグによってアノテーションされたデータとして見ることもできます。
- 例えば、オープンデータでは以下のようなメタデータがアノテーションされておりデータを検索、利用しやすくしています。
 - タイトル、組織名、作成者、タグ、公開・更新日、URL、データ形式、ファイルサイズ、使用言語、ライセンス
- 例えば、政府のデータカタログのメタデータは以下からダウンロードすることができます
 - <https://www.data.go.jp/for-developer/for-developer>

- DATA.GO.JP | 政府データカタログのメタデータ

アノテーション

- ・ ウェブ上の（ファイルからIoTの”モノ”に至るまでの）リソースのメタデータをアノテーションする枠組みとしてRDF（Resource Description Framework）があります。
- ・ オープンデータをRDFなどで記述し構造化した上で相互にリンクさせて活用するための枠組みとしてLOD（Linked Open Data）があります。
- ・ LODはウェブ上のデータを公開または利用する方式（公開されたデータそのものを指す場合もある）です。LODではデータの情報はRDFで記述され、データ間の関係を表すラベルが付与され、データ同士がリンクで結ばれたデータのウェブを形作るものです。
- ・ LODの例として、Wikipediaから抽出した情報をLODとして公開しているDBpediaがあります。
 - ・ DBpedia: <http://ja.dbpedia.org/>

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

36

- ・ DBpedia | Wikipediaから情報を抽出してLOD (Linked Open Data)として公開するコミュニティプロジェクト (CC-BY-SA 3.0)

アノテーション

DBpediaの「東京大学」に関するページの例

対象となるエンティティの情報はRDFで記述されている

The screenshot shows two DBpedia pages side-by-side:

- Left Page (University of Tokyo):** Shows the "About: 東京大学" page. A callout box highlights the "foundedBy" relationship, pointing to the "About: 国立大学法人" page.
- Right Page (Tokyo):** Shows the "About: 東京都" page. A callout box highlights the "locationCity" relationship, pointing to the "About: 東京都" page.

Both pages display RDF triples in the "Property" and "Value" columns, illustrating how entities are interconnected through these relationships.

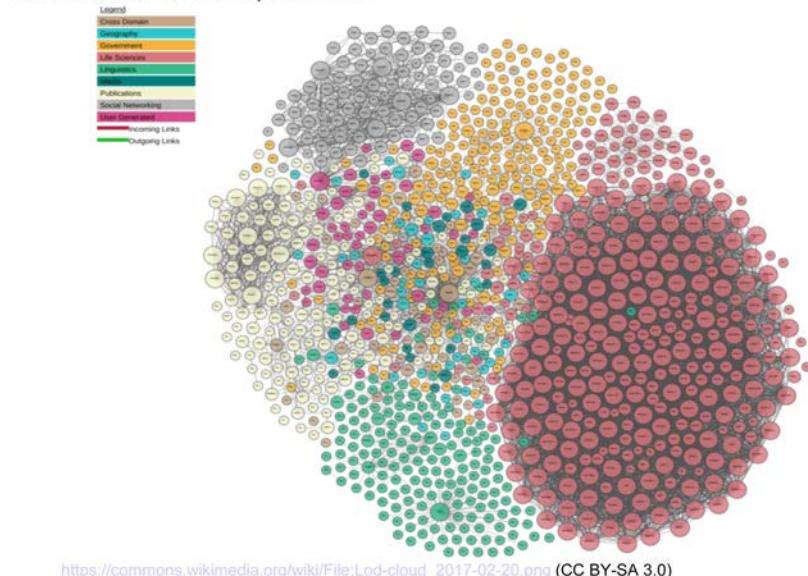
東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

37

- ・ DBpedia 信州大学に関するページ一覧のクエリ例 (CC-BY-SA 3.0)

アノテーション

さまざまなLinked Open Data



https://commons.wikimedia.org/wiki/File:Lod-cloud_2017-02-20.png (CC BY-SA 3.0)

東京大学 数理・情報教育研究センター 森純一郎 2021 CC BY-NC-SA

38

- The Linked Open Data Cloud (CC-BY-SA 3.0)

memo