第2章 データの種類と読み込み

各種データ形式の取り扱い方

データ形式

Pythonでのデータ形式

- Pythonには多くのデータ形式に対応する
- pandasは多くのデータ形式の読み書きが可能で、データ分析へ活用することが可能

代表的なデータ(ファイル形式)

- CSV形式
- Excel形式
- JSON形式
- HTML形式
- XML形式
- 文書データ
- 画像データ
- 音声データ
- RDBデータ
- pickle形式
- parquet形式

CSV形式

- CSVは「Comma Separated Values（カンマ区切りバリュー）」の略
- CSV形式は、カンマ（,）で列を区切り、改行コードで行を区切るデータ形式
- 2次元の表形式のデータをテキストで表す
- テキストファイルとして扱え汎用的で多くの場面で使われるデータ形式

```
"表章項目","人口","概算値","全国","時間軸（年月日現在）","年齢5歳階級","/男女別","男女計","男","女"
"人口【万人】","総人口","概算値","全国","2021年12月","総数","","12,547","6,099","6,448"
"人口【万人】","総人口","概算値","全国","2021年12月","0～4歳","","437","224","214"
"人口【万人】","総人口","概算値","全国","2021年12月","5～9歳","","503","258","245"
"人口【万人】","総人口","概算値","全国","2021年12月","10～14歳","","535","274","261"
"人口【万人】","総人口","概算値","全国","2021年12月","15～19歳","","559","287","272"
:
```

※1 データ
「政府統計の総合窓口 e-Stat」「人口推計」データ
https://www.e-stat.go.jp/stat-search/files?page=1&toukei=00200524&tstat=000000090001

CSV形式 - open()

- open()関数とfor文でデータを1行ずつ読み込む

```
with open("data/FEH_00200524_230205124938.csv", "r", encoding="cp932") as f:   ←①
        for i, row in enumerate(f):   ←②
            for col in row.strip().split(","):   ←③
                print(col, end="|")   ←④
            print()
            print("===")
            if i > 1:   ←⑤
                break
```

① open()関数のキーワード引数でencoding="cp932"と与えている
サンプルのCSVファイルの文字エンコーディングが、Python標準で使うUTF-8ではなく、CP932（Shift-JISの拡張）
なので文字エンコーディングを指定
② 読み込んだテキストファイルを1行ずつ取り出す
③ カンマ（,）で分割
④ パイプ|で区切って出力
⑤ CSVファイルの先頭3行を出力するように指定

出力

```
"表章項目"|"人口"|"概算値"|"全国"|"時間軸（年月日現在）"|"年齢5歳階級"|"/男女別"|"男女計"|"男"|"女"|
===
"人口【万人】"|"総人口"|"概算値"|"全国"|"2021年12月"|"総数"|""|"12|547"|"6|099"|"6|448"|
===
"人口【万人】"|"総人口"|"概算値"|"全国"|"2021年12月"|"0～4歳"|""|"437"|"224"|"214"|
===
```

文字コード

- 文字コードは、コンピュータが文字を正しく認識・処理するために欠かせない「文字と数字の変換ルール」
- コンピュータは、0と1の数字しかわからない
- 言語などにより複数の変換ルールがある
  - 例：
    - 日本語：「41」は「あ」
    - 英語：「41」は「A」
- 文字コードが違うと、正しい文字が表示されず、意味不明な記号になる
- 文字化けを防ぐためには、データの作成時と読み込み時で同じ文字コードを使用することが非常に重要

日本語の文字コードである「Shift_JIS (cp932)」と「UTF-8」

Shift_JIS (cp932)

- Shift_JISは主にWindowsやmacOSの**古いバージョン**での**日本語**の標準文字コード
- Windowsでは後方互換性の為にまだ有効となっている
- cp932（Code Page 932）：Shift_JISの中でMicrosoftが独自に拡張したもの
  - NEC特殊文字、IBM拡張文字、外字領域などが含まれており、Shift_JISとほぼ同じ意味で使われている
- CSV形式はMs-Excelのからの流れでShift_JISのデータが多い

UTF-8

- UTF-8は、現在インターネットや多くのシステムで**世界標準**となっている文字コード
- 世界中のほとんどの文字（アルファベット、日本語、中国語、アラビア語など）が扱える
- Shift_JISとは異なり、どの文字もバイト列の先頭から判別できる仕組みになっている
- UTF-8のファイルには、「BOM付き」と「BOMなし」の2種類がある
  - BOMは「Byte Order Mark（バイト・オーダー・マーク）」の略で、ファイルの先頭に付けらて「このファイルはUTF-8で書かれていますよ」という目印が付与される場合がある
  - プログラミングの世界では「BOMなし」と「BOM付き」のファイルもある

新しいウェブサイトやプログラムは今ではUTF-8が世界的に標準となっている

CSV形式 - pandas

- pandasでCSV形式のファイルを読み込むにはread_csv()関数を利用する
- 読み込んだデータはDataFrameになる

```
import pandas as pd

df = pd.read_csv("data/FEH_00200524_230205124938.csv", encoding="cp932")
```

データが適切に読めていることを確認するときに、head()メソッドやtail()メソッドがよく使う

```
df.head()
```

	表章項目	人口	概算値	全国	時間軸（年月日現在）	年齢5歳階級	/男女別	男女計	男	女
0	人口【万人】	総人口	概算値	全国	2021年12月	総数	NaN	12,547	6,099	6,448
1	人口【万人】	総人口	概算値	全国	2021年12月	0～4歳	NaN	437	224	214
2	人口【万人】	総人口	概算値	全国	2021年12月	5～9歳	NaN	503	258	245
3	人口【万人】	総人口	概算値	全国	2021年12月	10～14歳	NaN	535	274	261
4	人口【万人】	総人口	概算値	全国	2021年12月	15～19歳	NaN	559	287	272

データの行と列を確認

```
df.shape
```

```
(378, 10)
```

Excel形式

- Microsoft Excel (拡張子.xlsx)は表形式のデータをシートとし、複数のシートをブック形式で扱うデータ形式
- 内部的にはXMLファイルをZIP圧縮したファイル
- CSV形式のデータとは異なり、セルごとに色や表示形式などの書式を設定でき複雑な表現ができるデータ形式です。

```
import pandas as pd

df = pd.read_excel(“data/FEH_00200524_230205132438.xlsx”, header=12,  nrows=27, )
```

表2-4　read_excel()関数の主なキーワード引数

引数名	説明
sheet_name	シート名またはシートのインデックス番号を指定。リストで複数のシートを指定することも可能。デフォルトでは0となっているので、最初のシートが読み込まれる。
header	列名に使うヘッダ行を0から始まるインデックス番号で指定。デフォルトは0。
nrows	読み込む行数を指定。デフォルトはNoneで、すべての行を読み込む。
skiprows	読み込みをスキップする行の0から始まるインデックス番号のリストで指定、もしくは最初の何行を読み飛ばすかを整数で指定。デフォルトのNoneで、スキップしない。
index_col	DataFrameのIndexに使う列を指定。0から始まるインデックス番号で指定。デフォルトのNoneで、インデックス指定なし。
usecols	読み込む列を列名または列の0から始まるインデックス番号のリストを指定。デフォルトのNoneで、すべての列を読み込む。

JSON形式

- JSON形式のデータは、JavaScriptのオブジェクトを文字列表現したテキストファイル
- JavaScriptの文字列や数値を、配列やオブジェクトとして表現できる形式で、構造化したデータを扱える
- 階層があるツリー構造などをテキストファイルに書き出せる

JSONのデータをjsonライブラリのload()関数を使い読み込む
dict形式で読み込まれる

```
import json

with open("data/population-202302.json", "r") as f:
    data = json.load(f)
```

json_normalize関数：JSONデータをフラットなテーブル形式（DataFrame）に正規化する機能

JSONファイル

```
{
 "GET_STATS_DATA": {
   RESULT": {
       "STATUS": 0,
       "ERROR_MSG": "正常に終了しました。",
       "DATE": "2023-02-05T22:36:51.330+09:00"
     },
     "PARAMETER": {
       "LANG": "J",
       "STATS_DATA_ID": "0003443838",
       "DATA_FORMAT": "J",
       :
     }
  }
}
```

HTML形式

- HTML (Hyper Text Markup Language)形式とは 主にWebページ作成に用いられる
- 文字の装飾（色、サイズ、フォント）や、画像、動画などの配置して Webサイトを表現するために作られたフォーマット
- HTMLデータは、タグ(<>)で囲われたデータ構造のマークアップ言語でテキストファイル１である

HTMLファイル

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <title>ホームページ</title>
     :
  </head>
  <body>
    本文
     :
  </body>
</html>
```

XML形式

- XML（Extensible Markup Language）形式とは、データを構造化し、人間と機械の両方が理解できるテキスト形式で表現するマークアップ言語
- 定義されたタグ(<>)を用いてデータの意味や関係性を明確に示せる
- HTML形式とマークアップ言語という共通点があるが、目的が異なる
- HTMLは「ウェブページを美しく表示する」ための言語であり、XMLは「データを効率的にやり取りする」ための言語

XMLファイル

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
  <book id="bk101">
    <title>タイトル</title>
    <author>著者</author>
    <description>説明</description>
  </book>
</library>
```

文書データ

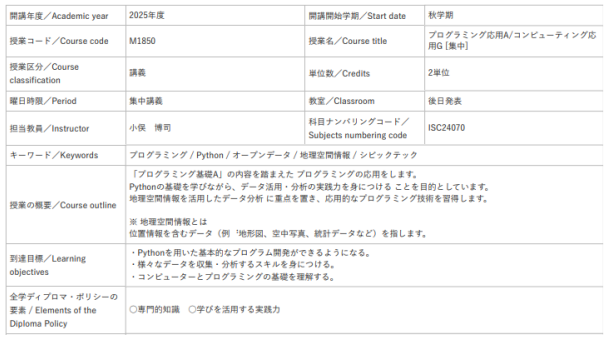
- 文書データは、人が読むための文章（テキスト）情報のファイル
- Microsoft WordやPDFのようなアプリケーションで作られたファイルは「バイナリ形式」である
- メモ帳のようなテキストエディタで読み書き可能なものを「テキスト形式」または「テキストファイル」と呼ぶ
- テキスト形式 (Text Format)は、人間が読める文字で構成されている
- バイナリ形式 (Binary Format)は、コンピュータが効率的に処理できるように数字データで構成されている。表示用のアプリケーションでしか表示できない

文書データ（テキスト形式）ファイル

購入リスト

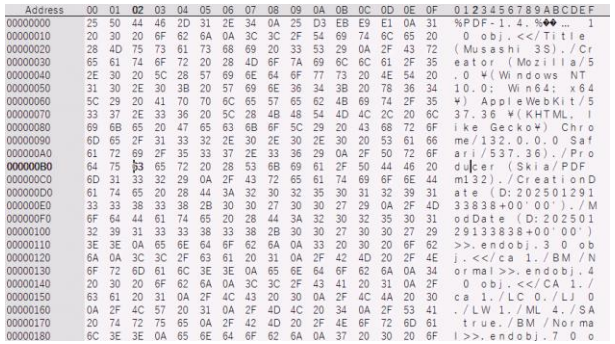
・牛乳
・パン
・卵
・りんご

文書データ（バイナリ形式）ファイル



開講年度／Academic year	2025年度	開講開始学期／Start date	秋学期
授業コード／Course code	M1850	授業名／Course title	プログラミング応用A/コンピューティング応用G [集中]
授業区分／Course classification	講義	単位数／Credits	2単位
曜日時限／Period	集中講義	教室／Classroom	後日発表
担当教員／Instructor	小俣　博司	科目ナンバリングコード／Subjects numbering code	ISC24070
キーワード／Keywords	プログラミング / Python / オープンデータ / 地理空間情報 / シビックテック		
授業の概要／Course outline	「プログラミング基礎A」の内容を踏まえた プログラミングの応用をします。 Pythonの基礎を学びながら、データ活用・分析の実践力を身につける ことを目的としています。 地理空間情報を活用したデータ分析 に重点を置き、応用的なプログラミング技術を習得します。 ※ 地理空間情報とは 位置情報を含むデータ（例 “地形図、空中写真、統計データなど）を指します。		
到達目標／Learning objectives	・Pythonを用いた基本的なプログラム開発ができるようになる。 ・様々なデータを収集・分析するスキルを身につける。 ・コンピューターとプログラミングの基礎を理解する。		
全学ディプロマ・ポリシーの要素 / Elements of the Diploma Policy	○専門的知識　○学びを活用する実践力		

アプリケーションで表示



Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	25	50	44	46	2D	31	2E	34	0A	25	D3	EB	E9	E1	0A	31	%PDF-1.4.%◆◆ … 1
00000010	20	30	20	6F	62	6A	0A	3C	3C	2F	54	69	74	6C	65	20	0 obj.<</Title
00000020	28	4D	75	73	61	73	68	69	20	33	53	29	0A	2F	43	72	(Musashi 3S)./Cr
00000030	65	61	74	6F	72	20	28	4D	6F	7A	69	6C	6C	61	2F	35	eator (Mozilla/5
00000040	2E	30	20	5C	28	57	69	6E	64	6F	77	73	20	4E	54	20	.0 ¥(Windows NT
00000050	31	30	2E	30	3B	20	57	69	6E	36	34	3B	20	78	36	34	10.0; Win64; x64
00000060	5C	29	20	41	70	70	6C	65	57	65	62	4B	69	74	2F	35	¥) AppleWebKit/5
00000070	33	37	2E	33	36	20	5C	28	4B	48	54	4D	4C	2C	20	6C	37.36 ¥(KHTML, l
00000080	69	6B	65	20	47	65	63	6B	6F	5C	29	20	43	68	72	6F	ike Gecko¥) Chro
00000090	6D	65	2F	31	33	32	2E	30	2E	30	2E	30	20	53	61	66	me/132.0.0.0 Saf
000000A0	61	72	69	2F	35	33	37	2E	33	36	29	0A	2F	50	72	6F	ari/537.36)./Pro
000000B0	64	75	63	65	72	20	28	53	6B	69	61	2F	50	44	46	20	ducer (Skia/PDF
000000C0	6D	31	33	32	29	0A	2F	43	72	65	61	74	69	6F	6E	44	m132)./CreationD
000000D0	61	74	65	20	28	44	3A	32	30	32	35	30	31	32	39	31	ate (D:20250129
000000E0	33	33	38	33	38	2B	30	30	27	30	30	27	29	0A	2F	4D	33838+00'00')./M
000000F0	6F	64	44	61	74	65	20	28	44	3A	32	30	32	35	30	31	odDate (D:20250
00000100	32	39	31	33	33	38	33	38	2B	30	30	27	30	30	27	29	29133838+00'00')
00000110	3E	3E	0A	65	6E	64	6F	62	6A	0A	33	20	30	20	6F	62	>>.endobj.3 0 ob
00000120	6A	0A	3C	3C	2F	63	61	20	31	0A	2F	42	4D	20	2F	4E	j.<</ca 1./BM /N
00000130	6F	72	6D	61	6C	3E	3E	0A	65	6E	64	6F	62	6A	0A	34	ormal>>.endobj.4
00000140	20	30	20	6F	62	6A	0A	3C	3C	2F	43	41	20	31	0A	2F	0 obj.<</CA 1./
00000150	63	61	20	31	0A	2F	4C	43	20	30	0A	2F	4C	4A	20	30	ca 1./LC 0./LJ 0
00000160	0A	2F	4C	57	20	31	0A	2F	4D	4C	20	34	0A	2F	53	41	./LW 1./ML 4./SA
00000170	20	74	72	75	65	0A	2F	42	4D	20	2F	4E	6F	72	6D	61	true./BM /Norma
00000180	6C	3E	3E	0A	65	6E	64	6F	62	6A	0A	37	20	30	20	6F	l>>.endobj.7 0 o

実際のファイルの内容

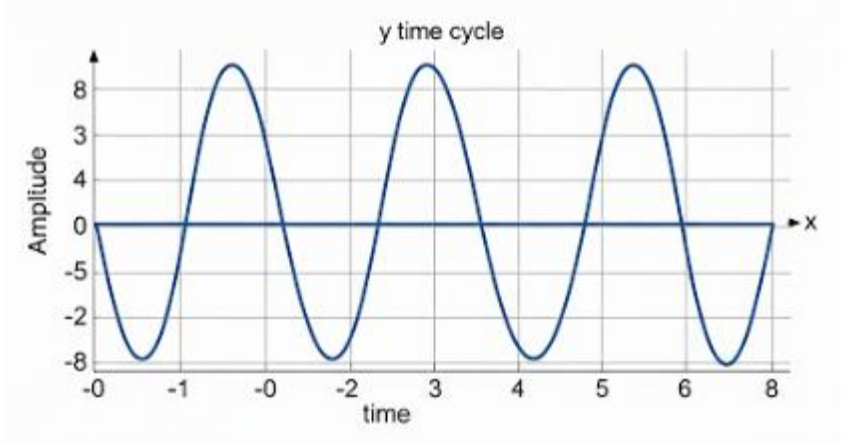
画像データ

- 画像データは、JPEGやPNGなどの画像のファイル形式
- バイナリ形式のファイル
- RGB（光の三原色：赤緑青）やCMYK（印刷用途：シアン、マゼンタ、イエロー、ブラック）などの色を表す数値や形状の情報を持っている

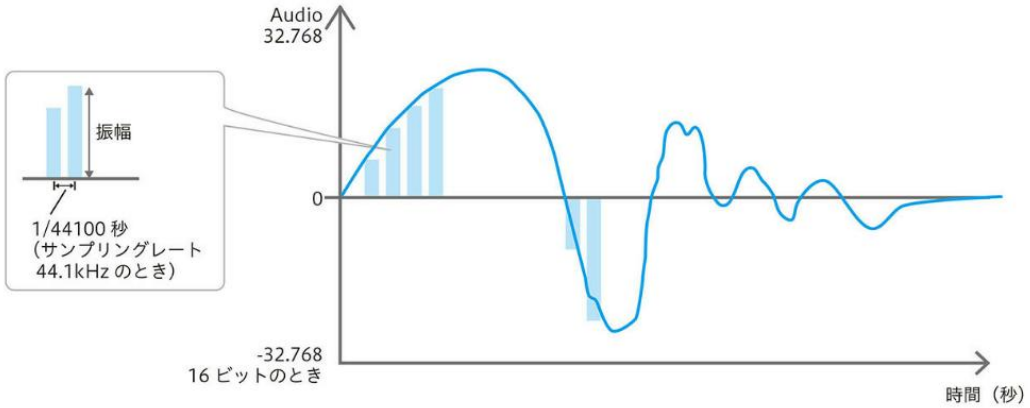
画像データ

音声データ

- 音声データとは、音をデジタル化して保存・再生するためのバイナリ形式のデータ
- 音楽やポッドキャスト、電話の録音、ボイスメモなどは音声データ
- 音のアナログな波をコンピュータで扱えるように、デジタル化として、「サンプリング（標本化）」し「量子化」したデータ



アナログ波形



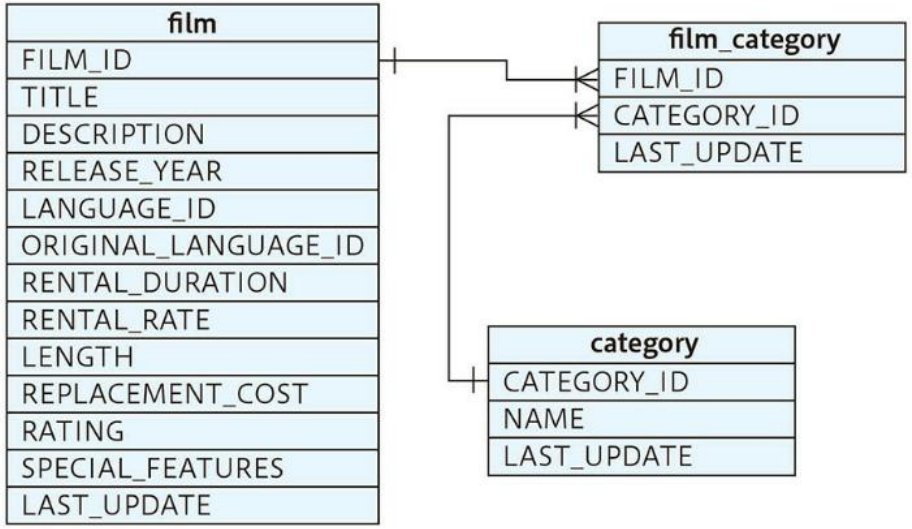
音声データ(デジタル化)

RDBデータ

- RDBデータ（リレーショナルデータベース(Relational Database)）は、データを「テーブル」と呼ばれる表形式のデータを複数のテーブルを関連付けて（リレーションさせて）データを扱うデータベースの形式
- バイナリ形式のデータ

	film_id	title	release_year	length	last_update	name
0	1	ACADEMY DINOSAUR	2006	86	2020-12-23 07:12:31	Documentary
1	2	ACE GOLDFINGER	2006	48	2020-12-23 07:12:31	Horror
2	3	ADAPTATION HOLES	2006	50	2020-12-23 07:12:31	Documentary
3	4	AFFAIR PREJUDICE	2006	117	2020-12-23 07:12:31	Horror
4	5	AFRICAN EGG	2006	130	2020-12-23 07:12:31	Family

テーブルデータ



テーブルデータの関係図(ER図)

pickle形式

- pickle形式は、Pythonのオブジェクトをそのままファイルに書き出せるPython専用の形式
- Pythonのオブジェクトをストレージに保存したり、データのやりとりに使える。
- Pythonオブジェクトそのものを保存
- バイナリ形式のデータ

parquet形式

- parquet形式は、Apache Parquetプロジェクトのデータ形式
- 表形式データを効率よく読み書きして高い圧縮率で保存できるデータ形式
- Python以外にも、JavaやC++やRなど多くのプログラミング言語で利用可能
- バイナリ形式のデータ
- pickle形式との違いは、pickle形式はPythonのオブジェクトをそのままバイナリ化したものでPythonでのみ利用可能
- parquet形式はデータ仕様に基づいて効率的にしたデータで汎用性がある

表2-5　pickle形式とparquet形式

	pickle 形式	parquet 形式
バイナリ化できるもの	すべてのPythonオブジェクト	表形式データのみ
圧縮	なし [※1]	あり
バージョンの影響	あり	なし [※2]

※1　gzipやbz2など、Pythonのほかのモジュールを使うことでデータを圧縮できます。
※2　parquet形式にはバージョン1と2がありますが、parquet形式をサポートするほとんどのライブラリがバージョン2に準拠していると考えられます。ただし、将来parquet形式のバージョンアップが行われると、バージョン違いによる不具合が起こる可能性があります。

Python ドキュメント

- Python公式ドキュメント
  - https://docs.python.org/ja/3/
- Python 言語リファレンス
  - Python 言語の文法について記述してある
  - https://docs.python.org/ja/3/reference/index.html
- Python 標準ライブラリ
  - Python とともに配付されている標準ライブラリについて説明してある
  - https://docs.python.org/ja/3/library/index.html

Pandas マニュアル

- Pandas
  - https://pandas.pydata.org/pandas-docs/stable/
  - User Guideは、Pandasの基本的な使用方法やデータ処理の手法についての詳細な解説がされている
  - User Guide
    - https://pandas.pydata.org/docs/user_guide/index.html
  - API ReferenceはPandasのクラスや関数に関する詳細な情報を提供している
  - API reference
    - https://pandas.pydata.org/docs/reference/index.html
- PyQ - pandas
  - https://docs.pyq.jp/python/pydata/pandas/index.html